P. 1877/74

12 1974

informatyka

SPIS TREŚCI NR 12/74

		str.
	Prognozowanie branży maszyn matematycznych — *Mieczysław Moszkowicz*	2
	Rekursja i iteracja w programowaniu — *Jowita Koncewicz*	5
	Programowanie heurystyczne — *Bogdan Stefanowicz*	8
	ACSES — zautomatyzowany system nauczania informatyki — *Jurg Nievergelt*	11
	O pojęciu i celu identyfikacji informacyjnej — *Lesław Aleksander Sieniawski*	15
	Optyczne pamięci cyfrowe — tłum. i oprac. *Jan Ryżko*	18
	Reguły prawidłowej eksploatacji i przechowywania pakietów dyskowych — oprac. *W. K.*	21
TRYBUNA CZYTELNIKA	Oznaczanie cyfry „0" i litery „O" — *Zygmunt Topolewski*	24
JS EMC	Operowanie danymi w DOS JS EMC — *Romuald Jagielski*	29
Z KRAJOWEGO BIURA INFORMATYKI I ZJEDNOCZENIA INFORMATYKI	Metoda szacowania czasu realizacji prac programowych — *Aleksander Kuźmin*	33
	Technika sieciowa w organizacji procesu projektowania systemów informatycznych — *Antoni Bielak*	35
	Standardy dokumentacji projektowej, programowej i eksploatacyjnej OBRI — *Henryk Zygier*	37
Z KRAJU	Starachowicki sposób na amortyzację komputera — *Krystyn Bernatowicz*	38
	Komputer w badaniach techniczno-eksploatacyjnych transportu samochodowego — *Benon Zb. Szałek*	41
ZE ŚWIATA	Stacja transmisji danych ICL 7503 — *Andrzej Bylicki, Zbigniew Stefański*	43
	We francuskim przemyśle chemicznym — *Maria Trzaskowska*	45
PROBLEMATYKA BAZY DANYCH	Metody reprezentacji i opisu struktur danych — *Władysław Bogucki, Witold Staniszkis*	48
PRZEGLĄD WYDAWNICTW	Recenzja: Urządzenia WE/WY — *Wojciech Kuczborski*	51
	Bibliografia wydawnictw polskich z dziedziny informatyki — oprac. *J. Klamborowski*	23 i 52
	Ogłoszenia	32, III i IV okł.

SPIS TREŚCI ROCZNIKA 1974 str. 25—28

WYDAWNICTWA CZASOPISM TECHNICZNYCH NOT
Warszawa
Czackiego 3/5

KOLEGIUM REDAKCYJNE

Redaktor Naczelny prof. dr hab. Leon ŁUKASZEWICZ

mgr Krystyn BERNATOWICZ, prof. dr hab. inż. Konrad FIAŁKOWSKI (zast. redaktora naczelnego), doc. dr inż. Zbigniew GACKOWSKI, mgr inż. Marek HOŁYŃSKI, Władysław KLEPACZ, Elżbieta KOŁODZIEJSKA, doc. dr hab. Antoni MAZURKIEWICZ, inż Dorota PRAWDZIC (zast. redaktora naczelnego)

Sekretarz Redakcji mgr Krystyna WROŃSKA — Red. tech. Józef DUSZA

RADA PROGRAMOWA

Przewodniczący — Prof. dr hab. Andrzej STRASZAK

Redakcja: 00-041 Warszawa, ul. Jasna 14/16, pokój 331, tel. 27-71-40 lub centrala 28-82-61 w. 285, dyżury redakcji 10,00—13,00

Zakład Kolportażu WCT NOT, Warszawa, ul. Mazowiecka 12

Zakł. Graf. „Tamka". Z. 2. Zam. 562. Papier druk. sat. IV kl. 70 g 61×86. Obj. 4 ark. druk. Nakład 6200+30. W-50

Cena egzemplarza zł 8.— INDEKS 36707 Prenumerata roczna zł 96.—

Informatyka

dawniej Maszyny Matematyczne

zastosowania w gospodarce, technice i nauce.

Nr 12
MIESIĘCZNIK
1974
ROK X
Grudzień

ORGAN KRAJOWEGO BIURA INFORMATYKI I POLSKIEGO KOMITETU AUTOMATYCZNEGO PRZETWARZANIA INFORMACJI NACZELNEJ ORGANIZACJI TECHNICZNEJ

P. 1877/74

Centralny Ośrodek Badawczo-Projektowy Górnictwa Odkrywkowego Wrocław

Rejestr czasop. Nr 744

23. GRU. 1974

Czytelnikom i Przyjaciołom naszego czasopisma najlepsze życzenia Świąteczne i Noworoczne

składa Redakcja

Mieczysław MOSZKOWICZ
Instytut Nauk Społecznych Politechniki Wrocławskiej

681.322.001.6(438)

Prognozowanie branży maszyn matematycznych

Autor porusza niektóre problemy prowadzenia studiów prognostycznych w odniesieniu do produkcji sprzętu informatyki w PRL, z uwzględnieniem celów branży, jej rozwoju i modelu funkcjonowania oraz wyboru horyzontu czasowego prognoz.

Obok nakreślenia ogólnej perspektywy rozwoju informatyki odrębnym torem powinno się prowadzić prace prognostyczne rozwoju branży maszyn matematycznych. Odrębność prognozowania branży — mimo wielu punktów zbieżnych z globalnym prognozowaniem rozwoju informatyki — wynika z zadań i funkcji branży jako kompleksu produkcyjnego, wkomponowanego w ogólnogospodarczy system zarządzania.

Zakładany wzrost produkcji i postępujące przemiany rynku informatycznego — wskazują, że branża maszyn matematycznych stanęła obecnie w obliczu poszukiwań nowej strategii swego rozwoju. Fakt ten jest tym bardziej godny podkreślenia, że z dniem 1.01.1973 r. do Zjednoczenia MERA, któremu organizacyjnie podlega branża, wprowadzono nowe zasady zarządzania [1]). Nie wnikając w ich szczegóły warto zaznaczyć, że stwarzają one instytucjonalne podstawy do regulowania rozwoju w oparciu o wyniki ekonomiczne.

Z opracowaniem strategii branży nierozerwalnie związane jest opracowanie prognoz jej rozwoju. Można wręcz stwierdzić, że od „jakości" prognoz zależeć będzie skuteczność przyjętej strategii.

Studia prognostyczne należy rozpatrywać wyłącznie z punktu widzenia pragmatycznego. Oznacza to, że prognozowanie będzie miało sens jedynie wówczas, jeżeli jego wyniki będą oddziaływały na proces podejmowania i przekazywania decyzji do realizacji. Wynika stąd, że opracowanie prognoz musi być dokonane przez pryzmat aktualnych problemów warunkujących prawidłowy rozwój branży. Dlatego też za punkt wyjścia procesu prognozowania należy przyjąć analizę tych problemów. One bowiem powinny wytyczać główne kierunki prognozowania, a w ślad za tym — kierunki działania.

Bariery rozwoju branży

Za podstawowe przeszkody przyspieszenia rozwoju produkcji sprzętu informatycznego można przyjąć:

- brak nowoczesnych podzespołów i elementów decydujących o najwyższych osiągnięciach techniczno-operacyjnych komputerów
- niska sprawność organizacyjna branży, wyrażająca się m. in. długim cyklem wdrażania wyrobów do produkcji — i wreszcie
- brak sprawnego mechanizmu badania i oddziaływania na potrzeby informatyczne.

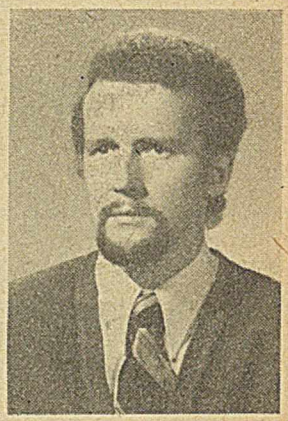
Pierwsza z wymienionych przeszkód ma pozornie charakter zewnętrzny, gdyż wynika z niedorozwoju przemysłów współpracujących przy wytwarzaniu sprzętu informatycznego. W rzeczywistości jednak jest to skutek wtórny niskiej — do niedawna — rangi przemysłu komputerowego w Polsce. Produkcja komputerów bowiem z charakterystyczną dla niej ścisłą symbiozą nauki z techniką, szybkim tempem zmian w technice i technologii, dużą liczbą nowych wyrobów, sama jest generatorem postępu technicznego i — jako taka — musi torować drogę dla tego postępu.

Efektywna i skuteczna realizacja tego zadania wymaga jednak wydatnego skrócenia okresu tworzenia nowej techniki komputerowej na rzecz zwiększenia okresu jej wykorzystania produkcyjnego. W przeciwnym przypadku, wyroby branży już w momencie wdrażania do produkcji są „skażone" zużyciem ekonomicznym. Skraca to okres ich życia rynkowego i, tym samym, obniża nowoczesność i konkurencyjność produkcji.

O ile dwa pierwsze z wymienionych problemów rozwoju przemysłu komputerowego są często sygnalizowane, to stosunkowo rzadko podnosi się problem badania i kształtowania potrzeb informatycznych. Rozpoznawanie potrzeb jest podstawowym wymaganiem kształtowania rozwoju każdej dziedziny produkcji. W przypadku branży maszyn matematycznych ma ono szczególne znaczenie i zasługuje na szerszą charakterystykę.

Przyjęło się określać cel omawianej branży jako zaspokojenie potrzeb sprzętowych w zakresie przetwarzania informacji. Pobieżna interpretacja tego sformułowania sugeruje, że celem branży jest wyprodukowanie — w przyjętym horyzoncie czasowym — określonej liczby komputerów. Takie rozumienie celu jest jednak nieścisłe. Użytkownikowi nie chodzi bowiem o posiadanie komputera jako takiego, lecz o jego funkcje użytkowe (użyteczność) w procesie przetwarzania informacji. Gdyby funkcje te były ściśle określone, jak np. lodówki, samochodu, roweru — cel branży bez trudu dałoby się sprowadzić do postaci ilościowej. W rzeczywistości jednak sprzęt informatyczny, w wyniku oddziaływania postępu technicznego i zmienności potrzeb podlega ciągłej ewolucji. Zmienia się również jego użyteczność, gdyż jest ona wyrazem pewnych relacji między własno-

[1]) Zob. Zasady działalności Zjednoczenia Przemysłu Automatyki i Aparatury Pomiarowej MERA, Warszawa, grudzień 1972 — (masz. powiel.).



Mgr MIECZYSŁAW MOSZKOWICZ ukończył Wydział Gospodarki Narodowej w WSE we Wrocławiu (1969 r.). Jednocześnie podjął pracę w Instytucie Nauk Społecznych Politechniki Wrocławskiej, zajmując się problemami prognozowania gospodarczego. Przygotowuje rozprawę doktorską na temat prognozowania rozwoju branży maszyn matematycznych.

ściami techniczno-operacyjnymi komputera a istniejącymi w jego zakresie potrzebami.

Z uwagi na to, cel branży należałoby uściślić jako maksymalizację funkcji użytkowych sprzętu informatycznego. Rozumiemy przez to zarówno rozszerzenie funkcji już obecnie spełnianych (poprzez wzrost liczby komputerów), jak i tworzenie nowych potrzeb informatycznych.

Postawienie wyżej sformułowanego celu zawiera dla branży maszyn matematycznych dość istotną implikację, a mianowicie: ilość i własności produkowanego sprzętu informatycznego powinny być wynikiem polaryzacji dwóch zmiennych — potrzeb oraz postępu naukowo-technicznego. Niezbędne przy tym jest, aby zmienna postępu naukowo-technicznego była permanentnie weryfikowana z punktu widzenia potrzeb. Brak tej weryfikacji bowiem zawiera niebezpieczeństwo utrzymania się dotychczasowego rozdźwięku między potrzebami w zakresie przetwarzania informacji a teoretycznymi możliwościami sprzętu informatycznego.

Jeśli się zważy rosnące wymagania użytkowników i związaną z nimi presję na poprawę ekonomiczności stosowanego sprzętu, oznaczałoby to pojawienie się trudności w realizacji produkowanych komputerów na rynku.

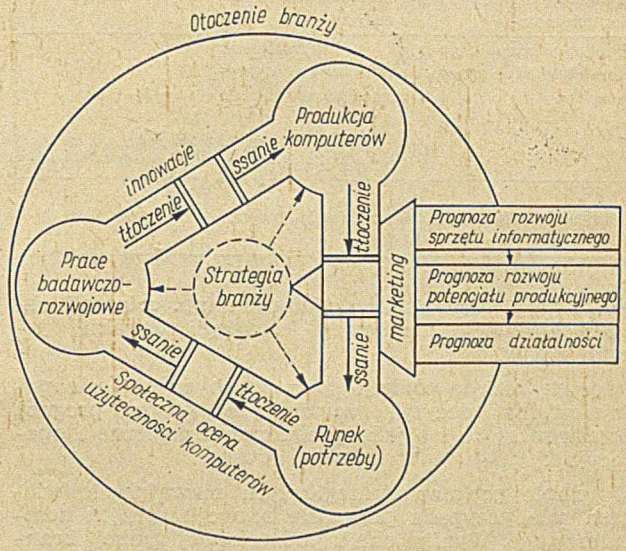
Dotychczasowe podejście branży do badania potrzeb informatycznych jest wynikiem określonych uwarunkowań społeczno-ekonomicznych. W szczególności zaś — wynikłym z systemu zarządzania — formalnym rozliczaniem podstawowych zadań (planu) oraz dużą chłonnością rynku na sprzęt informatyczny. W tej sytuacji funkcja zbytu w branży ograniczała się do czynności administracyjnych, związanych z rejestracją zamówień, przygotowywaniem wyrobów do wysyłki itp. Brak było natomiast zewnętrznych implikacji do badania rynku [2]) i stosowania związanych z nim strategii i nowoczesnych technik sprzedaży.

Aktualnie zaznacza się dość istotna ewolucja otoczenia branży. Jest ona związana zarówno z modyfikacją systemu zarządzania, jak i z ilościowymi i jakościowymi przemianami rynku komputerowego. Przyczyny tej ewolucji są dość złożone, dlatego pominiemy ich szersze omawianie. Warto jednak podkreślić, że w wyniku ich oddziaływania mamy obecnie do czynienia ze zjawiskiem względnego nasycenia rynku w sprzęt informatyczny. Wyrazem tego jest np. przesuwanie na dalszy termin dostaw zestawów komputerowych, bądź zupełna rezygnacja z ich zakupu [3]).

Nie ulega wątpliwości, że w istniejącej sytuacji warunkiem dalszego, bezkolizyjnego rozwoju branży jest operatywne i perspektywiczne badanie rynku oraz stworzenie ścisłej integracji między produkcją sprzętu informatycznego i potrzebami w zakresie przetwarzania informacji.

Model funkcjonowania branży — kierunki prognozowania

Z przytoczonych rozważań można wywnioskować, że z punktu widzenia optymalnego rozwoju produkcji branży niezbędne jest, aby sprzęt informatyczny był z jednej strony „ssany" przez odbiorców, z drugiej natomiast — „tłoczony" przez producenta. Podstawowym warunkiem istnienia takiej interakcji jest szybkie i zgodne z wymaganiami odbiorców wdrażanie innowacji technicznych. Z uwagi na duże tempo postępu technicznego w przemyśle komputerowym wymagania te są szczególnie istotne. Dlatego też obok mechanizmu „ssania" i „tłoczenia" w relacji produkcja-rynek, podobne mechanizmy pompy ssąco-tłoczącej powinny wystąpić w pozostałych sprzężeniach działalności branży, tj. prace badawczo-rozwojowe — produkcja oraz rynek — prace badawczo-rozwojowe (p. rys. 1) [4]).



Rys. 1. Miejsca prognoz w ogólnym modelu działalności branży maszyn matematycznych

Wypada zwrócić uwagę, że wyłączenie któregokolwiek elementu „ssania" bądź „tłoczenia" oznacza powstanie zaburzeń w rozwoju branży. Mogą się one przejawiać albo w postaci bezpośredniej obniżki efektywności działania branży (np. wystąpienie tzw. produkcji „na magazyn") albo jako niewykorzystanie wszystkich, potencjalnie istniejących możliwości rozwojowych (np. zbyt późne wdrożenie innowacji do produkcji).

Przeciwdziałanie wspomnianym zakłóceniom mieści się w ramach strategii rozwoju branży. Sprowadza się ono do zapewnienia korzystnej dla przyjętego celu stabilności między otoczeniem a wymienionymi sprzężeniami funkcjonowania branży. W trakcie działalności stabilność ta jest burzona przez występowanie dynamicznych (zewnętrznych i wewnętrznych) czynników rozwoju. Są nimi w szczególności postęp techniczny i zmienność potrzeb informatycznych. Dla sprawnego rozwoju branży i w związku z tym niezbędne jest możliwie dokładne prognozowanie zmian tych czynników w przyszłości. Ich antycypacja bowiem umożliwia ośrodkowi sterowania podjęcie w odpowiednim czasie działań mających na celu adaptację do niepomyślnych zakłóceń bądź ich likwidację.

Strategia branży musi zapewnić z jednej strony optymalny wybór przyszłych celów działania, z drugiej natomiast maksymalną precyzję w ustalaniu środków i metod realizacji tych celów. Adekwatnie do tych zadań strategii można wyróżnić pewne płaszczyzny prognozowania. Pierwsza z nich obejmuje swoim przekrojem wszelkie możliwe kierunki rozwoju branży i jako taka stanowi podstawę opracowania strategii rozwoju. Druga natomiast dotyczy prognozowania rozwoju niezbędnego — dla przyjętej strategii — potencjału produkcyjnego.

Odrębnym nurtem przebiegają prace prognostyczne mające na celu antycypację działalności realizowanej w ramach wybranego już kierunku strategii. Ten rodzaj prognoz stanowi podstawę do wyboru różne-

[2]) Jak stwierdza A. Hodoly „...administracyjne rozdzielnictwo towarów w warunkach trwałych ich niedoborów — charakteryzujące rynek sprzedawcy — czyni wszelkie badania rynku bezcelowe". — A. Hodoly — Wstęp do badań rynku, Warszawa 1961, s. 162—163.

[3]) Z rozmów z kierownictwem WZE MERA ELWRO.

[4]) Przy opracowywaniu schematu korzystano z pracy: Z. Kowal, Programowanie badań naukowych na podstawie teorii progów, Prace Naukoznawcze Politechniki Wrocławskiej, Wrocław 1971, s. 23.

Tabela

Horyzonty czasowe prognoz branży

Rodzaj prognozy	Cykle rozwoju naukowo-technicznego	Czas cyklu i horyzont prognozy
Prognoza rozwoju komputerów	Podstawowe odkrycia naukowe	15—20 lat
Prognoza rozwoju potencjału produkcyjnego	a) urzeczywistnienie podstawowego odkrycia (od planowania do realizacji) b) moralne zużycie środków trwałych	5—7 lat
Prognozy działalności	Moralne zużycie wyrobów	około 5 lat

go rodzaju strategii szczegółowych (np. srtategii obniżki kosztu jednostkowego, strategii wysokiej jakości itp.) [5].

Odpowiednio do funkcji spełnianych w procesie podejmowania decyzji, wymienione grupy prognoz określane są często jako prognozy strategiczne i operacyjne.

W grupie prognoz strategicznych niewątpliwie wiodąca rola przypada prognozie rozwoju sprzętu komputerowego. Wynika ona z podstawowej funkcji branży, jaką jest zaspokojenie potrzeb informatycznych. Zasadniczym celem tej prognozy jest określenie na płaszczyźnie ewolucji potrzeb i postępu technicznego — przypuszczalnych kierunków rozwoju technicznego komputerów. Nie chodzi tu bynajmniej o szczegółowe przewidywania parametrów technicznych sprzętu informatycznego. Takie zadanie — szczególnie przy długim horyzoncie czasowym — wydaje się niewykonalne. Głównym zamierzeniem prognozy rozwoju komputerów jest sprecyzowanie tych problemów, których rozwiązanie w przyszłości wymaga już obecnie określonych przedsięwzięć naukowo-badawczych.

Z prognozą rozwoju sprzętu informatycznego bezpośrednio koresponduje prognoza rozwoju potencjału produkcyjnego. Jej treść powinna określać dostosowane do przyjętych ustaleń w prognozie rozwoju komputerów wielkość i kierunki zabezpieczenia materialno-technicznego. I tak, w obszarze omawianej prognozy powinny znaleźć się: nakłady na prace naukowo-badawcze, nakłady na inwestycje produkcyjne i nieprodukcyjne, tendencje w rozwoju układu kooperacyjnego, zakup licencji, niezbędny import itp. Odmienny charakter mają prognozy operacyjne, które w ujęciu rodzajowym można nazwać umownie prognozami działalności.

O ile prognozy strategiczne określają generalne kierunki rozwoju branży, to zadanie prognoz operacyjnych polega na umożliwieniu wyboru „najlepszej drogi" w ramach przyjętego kierunku. Dlatego też obszar tego typu prognoz sprowadza się głównie do antycypacji działalności prowadzonej w ramach już zaangażowanego majątku produkcyjnego, a ich wykorzystanie ma miejsce w trakcie optymalizacji produkcji. W obliczu powyższego, przedmiotem prognoz działalności są przede wszystkim: wielkość i struktura popytu, asortyment i skala produkcji, koszty, akumulacja, wielkość rezerw itp.

Pewne kontrowersje budzić może zaliczenie do omawianej grupy, prognoz popytu na sprzęt informatyczny. Tym bardziej, że w wielu opracowaniach prognostycznych, w tym również dotyczących branży maszyn matematycznych [6], ilościowe prognozy popytu ujmowane są długofalowo, a więc wybiegają poza wykorzystanie istniejącego potencjału produkcyjnego. Wyjaśnienie tego zagadnienia wymaga uprzedniego naświetlenia problemu horyzontu czasowego prognozowania.

Horyzont prognozowania

Zagadnienie wyboru optymalnego horyzontu prognozowania nie jest teoretycznie rozwiązane. Być może z tego względu, większości opracowywanych prognoz nadaje się ramy czasowe, typowe dla naszej praktyki planowania wielookresowego, a więc 15, 5 i rzadziej 2—3 lat. Podejście to, poprawne z formalnego punktu widzenia, nie zawsze da się pogodzić z postulatem pragmatyczności prognoz. Wynika z niego bowiem, że zasięg czasowy prognoz powinien odpowiadać horyzontowi odnośnych decyzji, a tych nie da się ująć w sztywne ramy czasowe. Dla podejmującego decyzję ustalenie odpowiedniego horyzontu prognozowania jest o tyle istotne, że jego nieuzasadnione wydłużenie może spowodować spadek realności prognozy i w konsekwencji — nieskuteczność decyzji.

Długość horyzontu czasowego prognoz i decyzji jest wypadkową szeregu czynników, wśród których za najważniejszy można uznać specyfikę dziedziny, której one dotyczą. W branży maszyn matematycznych specyfika ta polega na ścisłej integracji nauki z produkcją i wynikających stąd: dużego tempa postępu technicznego i szybkiego starzenia się wyrobów i środków produkcji. Wynikające z tych zjawisk trudności prognozowania długofalowego skłaniają do szczególnie pieczołowitego wyboru długości okresu przewidywania.

Rozwój branży przebiega w postaci nawarstwiających się procesów naukowych, technicznych i rynkowych. Dlatego, jak się wydaje, najbardziej obiektywną podstawą wyboru horyzontu czasowego prognozowania branży są cykle rozwoju tych procesów [7].

Zaletą takiego ujęcia horyzontów czasowych jest to, że stwarza ono możliwość „zamknięcia" prognozy w ramach naturalnych etapów rozwoju prognozowania procesów, które wyznaczają jednocześnie zasięg czasowy podejmowanych decyzji. Umożliwia ono ponadto wybór horyzontu czasowego dostosowanego do rangi przedmiotu prognozy w kształtowaniu rozwoju branży.

Jak wiadomo, główny ciężar rozwoju informatyki przesuwa się obecnie ze sprzętu na oprogramowanie. Oznacza to, że również ranga samego wytwarzania sprzętu informatycznego względnie maleje na rzecz badań nad jego zastosowaniem. Dlatego decydujące znaczenie dla strategii branży będzie miało prognozowanie wyników tych badań i ich konsekwencji dla rozwoju technicznego komputerów. Oznacza to jednocześnie, dlaczego ilościowa prognoza popytu nie powinna wybiegać zbyt daleko w przyszłość. Wpływ badań nad zastosowaniami komputerów spowoduje bowiem niewątpliwie ewolucję ich funkcji użytkowych. Przy braku mierników tej ewolucji, każda prognoza zapotrzebowania musi być opracowana przez pryzmat aktualnych wyobrażeń o użyteczności sprzętu informatycznego. W tej sytuacji długofalowe prognozy zapotrzebowania komputerów mogą doprowadzić do zupełnie nierealnych wyników.

Przytoczone rozważania sygnalizują jedynie pewne problemy prognozowania przemysłu informatycznego. Jednak nawet na ich podstawie można stwierdzić, że jego prognozowanie nie może polegać na mechanistycznym ustalaniu przyszłych tendencji rozwojowych. Podstawowym warunkiem praktycznej przydatności prognoz jest bowiem ich dostosowanie do wymagań strategii zarządzania.

[5] Zob. szerzej: J. Kwejt, Metody i strategia zarządzania przedsiębiorstwem przemysłowym, Warszawa 1970, str. 225 i kolejne.

[6] Zob. np.: Prognoza normatywna branży informatyki, Faza II, Prognoza zapotrzebowania na wyroby i usługi, 23.III. 1973 r., opracowanie Przedsiębiorstwa Projektowania i Modernizacji Automatyki i Aparatury Pomiarowej MERAL.

[7] H. D. Hanstein, K. Neumann, Analiza ekonomiczna poziomu produkcji przemysłowej. Berlin 1965, s. 183 i kolejne.

Jowita KONCEWICZ
Instytut Maszyn Matematycznych
Warszawa

681.322.06

Rekursja i iteracja w programowaniu

Omówiono pojęcie rekursji i możliwości jej zastosowania do programowania. Podano różnice między procesami iteracyjnymi a rekursywnymi.

Pojęcie rekursji należy do tych zagadnień, które podczas nauki podstaw programowania sprawiają najwięcej kłopotów. Zdarza się też często, że nawet programiści dysponujący solidnym przygotowaniem zawodowym i dużym praktycznym doświadczeniem — z trudem odróżniają rekursję od iteracji. Często również można spotkać się ze zdaniami wyrażającymi zwątpienie w celowość stosowania rekursji w programowaniu. Nie będzie to tak bardzo zaskakujące, jeśli się zważy, że np. w obliczeniach numerycznych programy iteracyjne okazują się zazwyczaj najefektywniejsze, pomimo że opisy zastosowanych w nich algorytmów bywają dużo prostsze i bardziej eleganckie wówczas, gdy mają postać definicji rekursywnych.

W celu lepszego zrozumienia pojęcia rekursji, które pojawiło się w programowaniu jako odbicie stosowanych w matematyce definicji rekurencyjnych, przypomnijmy kilka pojęć z teorii funkcji obliczalnych. Zacznijmy od pojęcia indukcyjnej definicji funkcji za pomocą schematu rekursji prostej. Jest to najprostszy ze schematów indukcyjnych pozwalających określać pewne funkcje za pomocą innych funkcji. Jak każda definicja indukcyjna, schemat ten składa się z dwóch części, którymi są:

- warunki wyjściowe
- warunki indukcyjne.

Przypuśćmy, że mamy zdefiniowane dwie funkcje: jednoargumentową oraz trójargumentową h; powiemy, że dwuargumentowa funkcja f jest zdefiniowana przez rekursję prostą, jeśli są spełnione warunki:

$$f(0, x) = g(x) \tag{1}$$

$$f(n+1, x) = h\big(x, n, f(n, x)\big)$$

Zawężeniem tego schematu do funkcji jednoargumentowych będzie następujący schemat:

$$f(0) = k \tag{2}$$

$$f(n+1) = h(n, f(n)),$$

gdzie k oznacza pewną stałą. Przykładem zastosowania schematu (2) może być indukcyjna definicja funkcji silnia $n!$ za pomocą mnożenia.

$0! = 1$

$(n+1)! = (n+1) \cdot n!$

Schemat rekursji prostej (albo inaczej nazywanej rekursją pojedynczą) nie jest najogólniejszym schematem indukcyjnego określania funkcji w arytmetyce. W schemacie tym tylko jedna zmienna definiowanej funkcji zmienia się z n na $n+1$. Istnieją schematy równoczesnej rekursji ze względu na kilka zmiennych, nie dające się sprowadzić do schematu rekursji prostej. Schemat ten bowiem nie wystarcza do zdefiniowania wszystkich funkcji obliczalnych zmiennych naturalnych i przyjmujących wartości naturalne (umawiamy się, że liczba 0 należy do liczb naturalnych), ale jedynie klasy tzw. funkcji pierwotnie rekurencyjnych, będącej podklasą klasy funkcji obliczalnych (inaczej nazywanych też funkcjami ogólnie rekurencyjnymi).

Klasa funkcji pierwotnie rekurencyjnych jest najmniejszą klasą funkcji określonych na liczbach naturalnych i przyjmujących wartości naturalne, zawierającą funkcje

1. $S(x) = x + 1$ (następnik)
2. $Z(x) = 0$ (zero)
3. $I(x) = x$, $I(x, y) = y$ (rzutowanie zmiennych)

zamkniętą ze względu na operacje złożenia (superpozycji) funkcji i indukcji według schematu rekursji prostej.

Klasa funkcji pierwotnie rekurencyjnych obejmuje zatem te wszystkie funkcje, które można otrzymać za pomocą zastosowania skończonej liczby operacji złożenia funkcji oraz operacji rekursji prostej, wychodząc od funkcji zero, następnik i rzutowanie zmiennych.

Ażeby mieć aparat pozwalający uzyskać wszystkie funkcje obliczalne, trzeba do wymienionych sposobów określania funkcji dołączyć jeszcze jeden: definiowanie za pomocą operacji minimum. Operację tę stosuje się do dowolnej relacji $R(x, y)$ i otrzymuje w wyniku liczbę lub funkcję oznaczoną symbolem

$$(\mu x)\,[R(x, y)] \tag{4}$$

który rozumie się jako „najmniejsze takie x, dla którego zachodzi relacja $R(x,y)$". Mając dowolną relację $R(x, y)$ możemy za pomocą operacji minimum zdefiniować funkcję $f(y)$ następująco:

$$f(y) = (\mu x)\,[R(x, y)] \tag{5}$$

Na przykład część całkowitą pierwiastka z liczby y możemy określić wzorem:

$$E(\sqrt{y}) = (\mu x)\,[(x+1)^2 > y]$$

Operację minimum stosujemy do relacji $R(x, y)$ efektywnie wówczas, gdy jest spełniony waru-

Mgr JOWITA KONCEWICZ od chwili ukończenia studiów na wydziale matematyczno-fizycznym Uniwersytetu Warszawskiego pracuje w Instytucie Maszyn Matematycznych. Zajmuje się teorią i praktyką języków programowania maszyn cyfrowych. Pracowała przy realizacji kilku translatorów języków SAKO i ALGOL 60. Obecnie zajmuje się teoretycznym ujęciem zagadnienia translacji.

nek, że „dla każdej liczby naturalnej y istnieje taka liczba naturalna x, że zachodzi relacja $R(x, y)$".
W takim przypadku operację minimum nazywamy m i n i m u m e f e k t y w n y m. Klasa f u n k c j i obliczalnych jest najmniejszą klasą funkcji określonych na liczbach naturalnych i przyjmujących wartości naturalne, zawierającą zero, następnik i rzutowanie zmiennych, zamkniętą ze względu na operacje złożenia funkcji, rekursji prostej i minimum efektywnego.

Zauważmy, że pojęcie „rekursja", występujące w określeniach schematu rekursji oraz funkcji pierwotnie rekurencyjnych nie dotyczyło właściwie samych funkcji, ale tylko sposobu ich definiowania. Pojęcie to łatwo przenosi się do innych dziedzin, w których stosujemy maszyny cyfrowe. Spotykamy się tam z takimi przykładami rekursji, jak: rekursywne wywołania procedur, programy rekursywne, rekursywne przetwarzanie struktur listowych i napisów.

Rekursja a iteracja

W metodach numerycznych i w programowaniu spotykamy się często z pojęciem i t e r a c j i. W odróżnieniu od rekursji, pojęcie to nie ma właściwie znaczenia teoretycznego, ma natomiast dużą wartość praktyczną. Jeżeli algorytm obliczania wartości jakiejś funkcji potrafimy przekształcić z postaci rekursywnej do postaci iteracyjnej, to na ogół program realizujący ten algorytm będzie napisany w sposób bardziej skomplikowany, niż program dla definicji rekursywnej, ale (w przypadku większości maszyn cyfrowych) będzie dużo szybciej produkował wyniki, zatem korzystanie z niego będzie bardziej opłacalne. Zarówno iteracja, jak i rekursja opierają się na indukcji. Jednakże w przypadku rekursywnej definicji funkcji, wartość funkcji dla jakiegoś argumentu określamy w zależności od wartości innej funkcji, której argumentem może być znów inna wartość definiowanej funkcji dla innego jej argumentu. Ta inna wartość jest również określona w analogiczny, uzależniony sposób — po prostu jest określona tym samym wzorem, a więc nie jest znana *explicite*, chyba że jest nią wartość funkcji określona warunkiem początkowym.

Taki sposób definiowania powoduje jak gdyby „zapętlenie" definicji. Oczywiście w każdej poprawnej definicji rekursywnej musi występować jeden początkowy warunek nierekursywny, umożliwiający „wyjście z pętli". Na przykład w definicji (3) funkcji silnia jest to określenie wartości 0!

Definicja i t e r a c y j n a określa wartość funkcji w zależności od wielkości znanych już za każdym razem stosowania definicji. Obliczając zatem wartość jakiejś funkcji dla danego argumentu za pomocą algorytmu iteracyjnego, obliczamy kolejno jej wartości począwszy od jakiegoś argumentu początkowego; następnie po każdym wykonanym obliczeniu sprawdzamy, czy została już obliczona wartość funkcji dla zadanego argumentu. Jeżeli nie, to zwiększamy o pewną liczbę (k r o k i t e r a c j i) wartość argumentu i powtarzamy obliczanie dla tego nowego argumentu. Tak więc w przypadku iteracji wykonujemy — za każdym razem zastosowania wzoru iteracyjnego — pełne obliczenie wartości funkcji dla jakiegoś argumentu i powtarzamy cały proces: obliczanie, sprawdzenie — dopóty, dopóki nie wykonamy obliczenia dla zadanego argumentu.

W przeciwieństwie do iteracji, rekursja powoduje „samozagnieżdżenie" wykonywanego obliczenia. Zanim ukończy się obliczanie wartości funkcji dla zadanego argumentu, przystępuje się do obliczania wartości tej funkcji dla innego argumentu, ponieważ w definicji występuje ona jako jeden z argumentów funkcji zdefiniowanej wzorem rekursywnym. Wprawdzie i tym razem powtarzamy cały proces: obliczenie, sprawdzenie (czy doszliśmy już do warunku początkowego, określonego nierekursywnie), ale wszystkie te obliczenia przerywamy, pozostawiamy na jakiś czas nieukończone, ponieważ są zależne od jeszcze nie znanych wielkości. Dopiero wówczas, gdy uzyskamy tę wartość argumentu, dla której wartość funkcji była określona nierekursywnie, możemy kolejno „zakończyć" wszystkie rozpoczęte obliczenia. W wyniku „odgnieżdżania" obliczeń otrzymamy wtedy poszukiwaną wartość funkcji.

Mówiąc najbradziej ogólnie, można tak scharakteryzować różnicę między procesem iteracyjnym, a rekursywnym. P r o c e s i t e r a c y j n y polega na powtarzaniu określoną z góry jakimś warunkiem liczbę razy pewnej operacji czy też czynności, przy czym każde wykonanie tej operacji musi być ukończone, zanim przystąpi się do sprawdzania warunku i ewentualnego powtarzania wykonywania takiej samej operacji. W p r o c e s i e r e k u r s y w n y m natomiast, zanim zostanie zakończone wykonywanie danej operacji, przystępuje się do sprawdzania warunku, od którego uzależnione jest dalsze powtarzanie takiej samej operacji i — ewentualnie — do wykonywania tej operacji od początku. Można by powiedzieć, że powtarzaną operację wykonuje się w przypadku rekursji jako podprogram identycznej, nieukończonej operacji.

Istotę rekursji w zastosowaniu do programowania łatwo dostrzec analizując działanie dwóch różnych programów realizujących to samo zadanie dwoma rozmaitymi sposobami; jeden z nich będzie działał iteracyjnie, drugi — rekursywnie. Przypuśćmy, że chcemy obliczyć sumę wartości jakiejś funkcji $f(k)$ dla kolejnych wartości $k = 1,2...n$, tzn. obliczyć sumę $f(1)+...+f(n)$.

W pierwszym programie znajdą się następujące instrukcje:

suma: = 0;
for k: = 1 **step** 1 **until** n **do**
suma: = suma + $f(k)$;

Program ten wykona n razy dodawanie składnika $f(k)$ kolejno i niezależnie, a więc w sposób iteracyjny. Jeżeli jednak ten fragment programu zastąpimy procedurą suma (f, k), której treść będzie zawierała instrukcję:

suma (f, k): = **if** $k = 0$ **then** 0 **else** f(k) + suma $(f, k\text{-}1)$;

to otrzymamy program działający rekursywnie, ponieważ w procedurze suma występuje wywołanie jej samej. Chcąc obliczyć wartość suma (f, n) trzeba podczas wykonywania tego programu prowadzić dwa stosy w pamięci maszyny cyfrowej. Jeden z nich będzie przechowywał kolejno wartości $f(k)$, zanim będą mogły być dodane do kolejnych wartości suma $(f, k\text{-}1)$, dla których obliczenia znów odwołujemy się do tej samej procedury. Drugi stos jest potrzebny do przechowywania kolejnych łączników do wywoływanej procedury, a właściwie „powieleń" tego samego odwołania. Dopiero po uzyskaniu wartości $k = 0$ będzie można wykonać wszystkie zapoczątkowane dodawania, likwidując równocześnie oba stosy.

Jak więc widzimy programy rekursywne wymagają podczas ich wykonywania tworzenia s t o s ó w przechowujących częściowe wyniki rozpoczętych operacji oraz kolejne „powielenia" argumentów i miejsc roboczych wywoływanych procedur. Dlatego w tych maszynach cyfrowych, które nie mają odpowiedniego aparatu do automatycznej realizacji stosu, programy rekursywne wykonują się znacznie wolniej od iteracyjnych, a przy tym wymagają zarezerwowania większej liczby miejsc roboczych w pamięci maszyny. Mają natomiast inną przewagę, są krótsze, łatwiejsze do zrozumienia, bardziej czytelne.

W omówionym powyżej przykładzie zastąpienie dodawania iteracyjnego procedurą rekursywną mogło wyglądać nieco sztucznie; program rekursywny nie był ani krótszy, ani prostszy od iteracyjnego, jego wykonanie zaś okazało się nadspodziewanie skomplikowane. Pokażemy teraz inne przykłady programów iteracyjnych i rekursywnych, w których ujrzymy wyraźnie pożytek wynikający z użycia w programie definicji rekursywnych.

Wróćmy zatem do przytoczonej na początku funkcji silnia. Definicję (3) można też tak sformułować:

$$\text{jeżeli } n = 0 \text{ to } n! = 1, \quad \text{inaczej } n! = n \cdot (n-1)! \qquad (6)$$

W celu obliczania wartości funkcji silnia można napisać zarówno program iteracyjny, jak i rekursywny. Porównajmy dwa programy w ALGOLU 60:

```
integer procedure silnia (n);
value n; integer n;
begin real f; integer i;
        f := 1;
        for i := 1 step 1 until n do                    (7)
            f := f × i;
        silnia := f
end;
```

```
integer procedure silnia (n);
value n; integer n;
silnia := if n = 0 then 1| else n × silnia (n — 1);   (8)
```

Zauważmy, że program (8) powstaje nieomal bezpośrednio z definicji funkcji silnia, zwłaszcza jeżeli posłużymy się jej postacią (6). Nie można tego powiedzieć o iteracyjnej procedurze (7).

Jako następny przykład weźmy algorytm znajdowania największego wspólnego dzielnika dwu liczba naturalnych. Wyraża go się wzorem:

$$\begin{aligned} \mathrm{NWD}(n, m) &= \mathrm{NWD}(m, n) && \text{gdy } m > n \\ &\ n && \text{gdy } m = 0 \qquad (9) \\ &\ \mathrm{NWD}(m, R(n, m)) && \text{gdy } m \leqslant n \end{aligned}$$

gdzie $R(n, m)$ oznacza resztę z dzielenia n przez m. Na podstawie tej definicji można natychmiast napisać procedurę rekursywną:

```
integer procedure NWD (n, m);
value n, m; integer n, m;
NWD: = if m>n then NWD (m, n) else
if m = 0 then n else NWD (m, R(n, m))                 (10)
```

Ułożenie programu iteracyjnego dla tego algorytmu będzie już wymagało nieco trudu, wprowadzenia dodatkowych zmiennych i etykiety.

Otrzymana procedura stanie się też znacznie mniej zrozumiała w przeciwieństwie do bardzo przejrzystej procedury (10).

```
integer procedure NWD (n, m);
        value n, m; integer n, m
        begin integer n1, m1, p;
if n>m, then begin n1: = n; m1: |= m end
        else begin n1: = m; m1: = n end;
L: if R (n1, m1) = 0 then NWD: = m1                   (11)
else begin
        p: = m1;  m1: = R (n1, m1);
n1: = p; go to L
end;
end;
```

Ponieważ nie istnieje ogólny skończony algorytm pozwalający każdą definicję rekursywną przekształcić w definicję iteracyjną, napisanie programu iteracyjnego realizującego algorytm rekursywny może częstokroć wymagać pewnej inwencji i sporo pracy. Z drugiej strony, programy rekursywne są zazwyczaj kłopotliwsze podczas uruchamiania, a ponadto działają na ogół dużo wolniej od iteracyjnych. Toteż wydaje się, że wszędzie tam, gdzie można łatwo proces rekursywny zastąpić iteracyjnym, nie należy z tego rezygnować. Dotyczy to zwłaszcza większości obliczeń numerycznych. Jednakże wszelkie programy przeznaczone do analizowania i przekształcania struktur, które mogą być zdefiniowane rekursywnie (odnosi się to przede wszystkim do działania na strukturach listowych, przekształcania napisów, analizy składni języków, wielu zagadnień z dziedziny „sztucznej inteligencji") okazują się efektywniejsze wówczas, gdy same są napisane rekursywnie. Wynika stąd też, że języki programowania przeznaczone do przetwarzania napisów lub do pisania translatorów powinny umożliwiać stosowanie rekursji.

LITERATURA

[1] Barron, D. W.: Recursive techniques in programming. Macdonald, London, 1968.

[2] McCarthy, J.: Towards a mathematical science of computation. Information processing 1962, North Holland, 1963.

[3] Grzegorczyk A.: Zarys logiki matematycznej. PWN, Warszawa, 1961.

[4] Higman B.: A comparative study of programming languages. Macdonald, London, 1967.

Bogdan STEFANOWICZ
Szkoła Główna Planowania i Statystyki
Warszawa

681.322.06:65.012.122

Zdjęcie i życiorys Autora publikowaliśmy w nr 3/74

Programowanie heurystyczne

Przedstawiono stan badań nad zastosowaniem metody heurystycznej do rozwiązywania problemów programowania komputerowego.

Analiza sposobów rozwiązywania problemów pozwala stwierdzić, iż istnieją dwie — zasadniczo różne — metody postępowania.

Pierwsza z nich polega na przeglądaniu wszystkich znanych i nasuwających się wariantów i sprawdzaniu, który z nich może zostać uznany jako poszukiwane rozwiązanie. Metoda ta bezpośrednio prowadzi do budowy odpowiednich algorytmów i gwarantuje, że w trakcie poszukiwania nie zostanie pominięte żadne rozwiązanie, a każde z nich może być dokładnie przeanalizowne. Ma ona jednak tę istotną wadę, że pochłania niekiedy bardzo dużo czasu, a niekiedy z tego powodu wręcz przestaje być w praktyce przydatna. Na przykład przy grze w szachy w celu wybrania właściwego ruchu na podstawie analizy wszystkich możliwych posunięć należałoby przejrzeć około 10^{120} rozwiązań. Analiza takiej liczby wariantów jest praktycznie niemożliwa. Toteż stosowanie w tym przypadku systematycznego, albo inaczej — algorytmicznego przeszukiwania przestrzeni możliwych rozwiązań praktycznie nie doprowadziłoby do żadnego wyniku [1]).

Inną metodą postępowania jest zrezygnowanie z pełnego ,systematycznego przeglądania możliwych wariantów i skoncentrowanie się od razu na tych z nich, które w danej sytuacji doświadczony specjalista rozważyłby w pierwszej kolejności. Przykładem może znów służyć gra w szachy, podczas której gracz ogranicza się do rozważania tylko niewielkiej liczby posunięć.

Metoda ta zwana heurystyczną, posiada tę istotną wadę, iż nie gwarantuje, że wśród pominiętych wariantów nie znajdują się właśnie te, które powinny być rozważone jako pierwsze. Nie gwarantuje ona nawet, że zostanie znalezione jakiekolwiek dopuszczalne rozwiązanie. Jej zaletą jest jednak to — że w przypadku trafnego wyboru wariantu rozwiązania, to ostatnie jest osiągane szybciej niż przy stosowaniu algorytmicznego podejścia do tego samego zagadnienia. Teoretycznie, algorytmiczne poszukiwanie odpowiedniego ruchu przy grze w szachy pozwala na znalezienie posunięcia najkorzystniejszego lecz praktycznie metoda ta nie doprowadziłaby do zakończenia nawet jednej partii ze względu na czas, jaki gracze musieliby zużyć na przeanalizowanie chociażby jednej sytuacji. Metoda zaś heurystyczna nie gwarantuje znalezienia posunięcia najlepszego (chociaż tego nie wyklucza) lecz pozwala w rozsądnym czasie wybrać posunięcie zadowalające.

[1]) Używany w tym kontekście termin „metody algorytmiczne" jest w rzeczywistości terminem ogólnym, odnoszącym się nie tylko do programów maszynowych, lecz także do każdego jednoznacznie sformułowanego postępowania „maszynopodobnego", które przy dokładnym stosowaniu się do niego zawsze prowadzi do osiągnięcia zamierzonego rezultatu.

Ta krótka charakterystyka metody heurystycznej pozwala dostrzec w niej cenne narzędzie przydatne przy komputerowym rozwiązywaniu wielu problemów, z którymi człowiek spotyka się w swej codziennej praktyce.

Prace i badania nad wykorzystaniem metody heurystycznej w programowaniu komputerów datują się od roku 1957, kiedy to A. Newell, J. C. Shaw i H. A. Simon opublikowali opis programu rozwiązującego pewne problemy z dziedziny logiki matematycznej. Programowi temu nadano nazwę „Teoretyk Logiki". Wywołał on duże zainteresowanie głównie dlatego, że przy jego opracowaniu wykorzystano metody empiryczne oraz reguły przyjęte na podstawie analizy procesu rozwiązywania podobnych zadań przez człowieka.

W niedługim czasie na podobny temat ukazała się praca H. Gelerntera (1959). Uczony ten przedstawił opis programu, w oparciu o który maszyna elektroniczna była w stanie dowodzić twierdzenia z zakresu geometrii euklidesowej. Jak stwierdza H. Gelernter, udowodnienie zwykłego dziesięciokrokowego twierdzenia geometrycznego na drodze wyczerpującego wytwarzania sekwencji dowodowych, zajęłoby każdej istniejącej maszynie cyfrowej czas rzędu tysięcy lat. Jest to w dużym stopniu prawdziwe nawet w chwili obecnej, kiedy to dysponujemy komputerami o wiele szybszymi niż te, które istniały w okresie pojawienia się pracy H. Gelerntera. Toteż opracowanie tego programu, podobnie jak i „Teoretyka Logiki", wymagało zastosowania heurystycznej metody programowania.

Od tego czasu na temat metody heurystycznej ukazało się sporo publikacji opisujących różne interesujące badania. Na szczególną uwagę zasługuje praca wymienionych już trzech uczonych amerykańskich, opisująca program pod nazwą „Rozwiązywacz Problemów Ogólnych [2]) (A. Newell, J. C. Shaw, H. A. Simon 1959).

Sam termin „heurystyczny" stosowany jest w dwojakim sensie. Po pierwsze — „heurystyczną" można nazwać taką czynność intelektualną człowieka, która prowadzi do rozwiązania skomplikowanego, nietypowego zadania. Po drugie — „heurystycznymi" zwane są te swoiste chwyty, które człowiek przyswaja sobie w toku rozwiązywania pewnych zadań i które — mniej lub bardziej świadomie — przenosi na inne zadania.

Chwyty tego rodzaju zwane są często po prostu heurystykami. „W ujęciu wielu autorów, heurystyki to reguły, zasady, wskazówki, taktyki, triki i intuicje, które regulują przebieg procesu poszukiwania rozwiązania, lecz nie gwarantują osiągnięcia pożądanego wyniku" (Kozielecki 1968, str. 38). Heurystyki zwane też są przez niektórych autorów strategiami (Feigenbaum i Feldman 1963, str. 157).

Badania w dziedzinie programowania heurystycznego doprowadziły do pewnej klasyfikacji heurystyk. Jednym z kryteriów takiej klasyfikacji jest uniwer-

[2]) termin angielski: General Problem Solver.

salność heurystyk. Z tego punktu widzenia wyróżnia się heurystyki ogólne i specjalne, zwane też specyficznymi. Do pierwszych należą te z nich, które przydają się przy rozwiązywaniu wielu różnych problemów. Przykładami tego typu heurystyk są następujące „dobre rady":

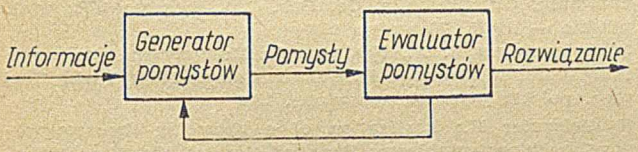
- problem złożony podziel na podproblemy i staraj się rozwiązać każdy z podproblemów z osobna
- nie upieraj się przy jednym punkcie widzenia, jeżeli to nie prowadzi do rozwiązania pomimo wykonanych wielu prób
- próbuj znaleźć analogię pomiędzy danym zagadnieniem, a zagadnieniami poprzednio rozwiązanymi.

Listę tego typu heurystyk można znacznie wydłużyć. Sporo przykładów podaje R. Ledley (162), wiele z nich jest także u J. Kozieleckiego (1968) i B. Puszkina (1967).

W praktyce przydatność heurystyk ogólnych jest raczej ograniczona. Większą rolę odgrywają heurystyki specjalne. Wiążą się one z określoną dziedziną i mogą być uznane jako „dobre rady" tylko przy rozwiązywaniu zagadnień pochodzących z tej dziedziny. Na przykład przy rozwiązywaniu problemów gry w szachy ważną heurystyką jest ta która sugeruje konieczność obrony własnego króla i atakowanie króla przeciwnika, natomiast przy opracowywaniu zasad weryfikacji danych rzeczą istotną jest ustalenie par cech współzależnych, ustalenie tzw. cech bazowych itd.

Z punktu widzenia treści heurystyki dzieli się na składniowe (strukturalne) i semantyczne. Nazwy te związane są z typem rozpatrywanego obiektu (analizowanej strony zagadnienia). I tak pierwsza z nich ma za zadanie usprawnić analizę badanej struktury, którą może być np. wzór matematyczny lub zdanie przy tłumaczeniu tekstów z jednego języka na inny. Heurystyki zaś semantyczne usprawniają zrozumienie sensu (treści) zawartej w analizowanej strukturze.

Reguły heurystyczne — czyli heurystyki — mają pewne cechy charakterystyczne. Najważniejszą z nich jest zawodność. Nawet najlepsze reguły tego rodzaju nie gwarantują, że problem zostanie rozwiązany. Są to więc metody, z którymi wiąże się pewne ryzyko niepowodzenia. Reguły heurystyczne różnią się między sobą stopniem skuteczności czy racjonalności. Jednakże współczesna wiedza na ten temat nie podaje żadnych metod, które pozwoliłyby stwierdzić wyższość jednych heurystyk nad innymi w określonych sytuacjach problemowych.



Następną cechą charakterystyczną reguł heurystycznych jest ich niepełna określoność. Nawet najlepiej sformułowane przepisy typu heurystycznego nie są podane w postaci ściśle określonej listy prostych operacji myślowych. Trzeba jednak za J. Kozielackim stwierdzić, że specjaliści zajmujący się programowaniem heurystycznym coraz częściej konstruują heurystyki, które ze względu na stopień precyzji przypominają algorytmy. Wynika to z konieczności dostosowania się do sposobu działania maszyn elektronicznych. Jak bowiem wiadomo, każde zadanie przekazywane komputerowi wymaga ścisłego sformułowania problemu i opisania go w odpowiednim języku programowania. Niestety, żaden z takich istniejących obecnie języków nie pozwala na proste posługiwanie się metodami heurystycznymi.

Stąd też budowa programu heurystycznego pociąga za sobą konieczność przedstawienia całego rozwiązania w sposób jednoznaczny, a więc w postaci algorytmu. W rezultacie więc programy heurystyczne na pierwszy rzut oka nie różnią się od programów opartych na algorytmach. Różnicy nie ma jednak tylko przy interpretowaniu obu typów programów jako zbioru reguł (rozkazów) dla komputera. Zarysowuje się ona natomiast na poziomie ogólnej strategii stosowanej w programie, która może być bądź strategią algorytmiczną, bądź też heurystyczną. Jak stwierdza M. J. Apter (1970), przypuszczalnie w ten sam sposób pracuje mózg ludzki: funkcjonowanie mózgu na poziomie neuronów musi być w pewnym sensie algorytmiczne, pomimo że na poziomie wyższym prawdopodobnie posługuje się on heurystycznymi technikami myślenia.

Jak zauważono, heurystyki w zasadzie nie gwarantują znalezienia rozwiązania. Stąd też stawia się im pewne wymagania dotyczące „opłacalności". Heurystyka — jako swoisty filtr — musi być wystarczająco „nieprzepuszczalna", aby w rezultacie zysk w efektywności rozwiązywania wzrastał. Po drugie, heurystyka będzie zazwyczaj eliminować ze zbioru rozpatrywanych wariantów pewną liczbę eleganckich i szybkich rozwiązań.

Wobec tego filtr ten musi być starannie dopasowany do tej podklasy problemów, które uważa się za interesujące i które zapewne będą dane maszynie do rozwiązania. Dla danej klasy heurystyk równowaga między tymi zasadniczo przeciwstawnymi wymaganiami jest w dużym stopniu funkcją organizacji i mocy obliczeniowej maszyny i może być krytyczną w pewnych warunkach (raczej łatwo osiągalnych).

Pomimo zawodności heurystyk i ryzyka związanego ze stosowaniem tego typu reguł przy rozwiązywaniu problemów empirycznie, stwierdzono ich dużą przydatność. Oto jak ocenia H. Gelernter, autor programu herystycznego dowodzącego twierdzenia z zakresu geometrii: „wydaje się... że maszyna do geometrii (tzn. program dowodzący twierdzenia z tego zakresu) daje przekonywujące świadectwo potęgi i płodności heurystycznego programowania w rozwiązywaniu problemów pewnej klasy przez maszynę cyfrową. W naszym doświadczeniu często dodanie pojedyńczej heurystyki powiększa moc dowodzenia twierdzeń maszyny w stopniu równoważnym trzy do pięciokrotnemu wzrostowi szybkości lub pojemności pamięci maszyny cyfrowej" (Feigenbaum i Feldman, 1963, wyd. polskie 1972, str. 170).

F. Wyman (1973) potwierdza te wywody wymownymi argumentami. Mianowicie napisany przez niego program, zasadzający się na heurystycznym szacowaniu pewnych parametrów w modelu regresji liniowej, zużywa zaledwie 10—15% czasu potrzebnego na oszacowanie tychże parametrów metodą algorytmiczną przy niemalże tej samej dokładności (program heurystyczny daje parametry obarczone błędem względnym nie przekraczającym 1—4% wartości uzyskiwanych w oparciu o algorytmy optymalizacyjne).

Programy heurystyczne, jak powiada B. Puszkin (1967), noszą na sobie piętno pewnej określonej koncepcji myślenia, którą nazywa on koncepcją labiryntu. Wedle tej koncepcji każde zadanie może być przedstawione w postaci labiryntu, którego wejście i wyjście reprezentują odpowiednie zbiory danych. Wejściem jest sytuacja problemowa, zaś wyjściem — sytuacja, którą należy uzyskać. Rozwiązanie zadania polega na znalezieniu drogi prowadzącej od wejścia do wyjścia.

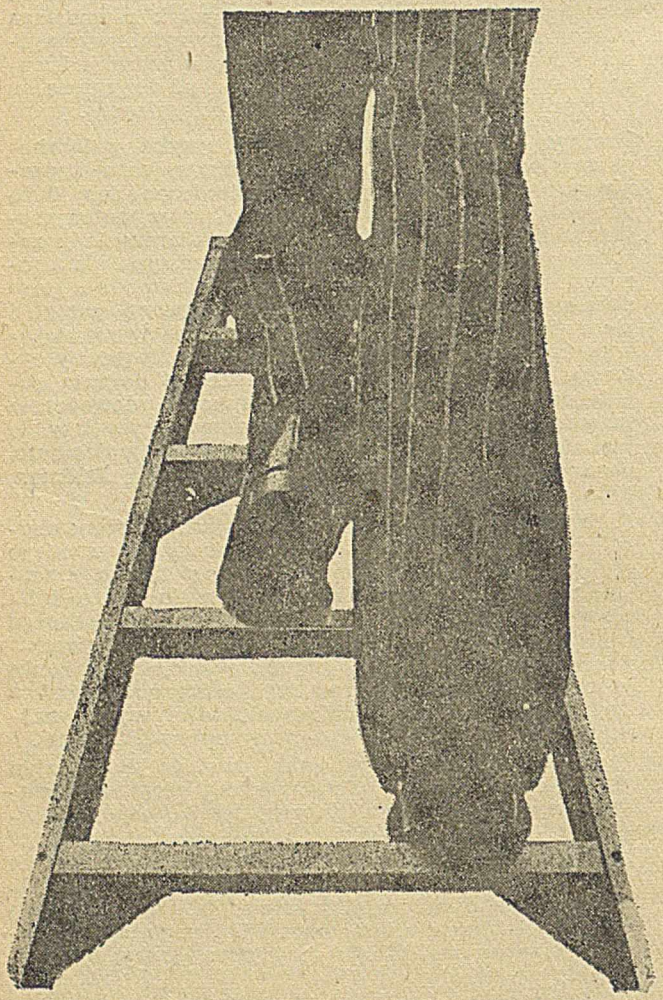
Drogę tę Slagle i R. S. Ledley usiłują znaleźć posługując się tzw. drzewem problemowym lub inaczej grafem. Budowa takiego grafu w trakcie rozwiązywania problemu przebiega w sposób następujący. Po określeniu ostatecznego celu następuje wyodrębnienie celów pośrednich (tzw. podcelów) oraz sposobów ich osiągnięcia. Na przykład niech ostatecznym celem będzie pewien cel G_0, to G_1 nazywa się podcelem.

Ogólnie, jeżeli G_{i-1} może zostać wyprowadzone z G_i, to G_i jest podcelem i-tego rzędu oraz wszystkie G_j dla $j > i$ nazywają się podcelami wyższego rzędu niż G_i.

Generowanie celów oraz sposobów przechodzenia od jednego celu do innego następuje w programie, którego ogólną strukturę interesująco ujął J. Kozielecki (1968).

Najogólniej można ją przedstawić jak na rysunku na str. 9.

Do generatora dopływają informacje ze świata zewnętrznego oraz z pamięci. Na ich podstawie generator próbuje wytworzyć pomysł rozwiązania. Jeżeli jest to niemożliwe, wówczas pobiera dodatkowe informacje z dostępnych mu źródeł. Wysunięty pomysł zostaje przesłany do ewaluatora, który ocenia jego przydatność. W przypadku oceny pozytywnej następuje zakończenie czynności rozwiązywania. W przeciwnym razie ewaluator przesyła do generatora odpowiedni sygnał i cykl pracy programu powtarza się. Przedstawiona struktura programu w dużym stopniu zależy od rodzaju rozwiązywanego problemu. I tak występują problemy, w których brak jest informacji na temat już istniejących możliwych rozwiązań. W takim przypadku na początku należy wytworzyć odpowiednie rozwiązanie, co czyni generator pomysłów, a dopiero potem można je wykorzystać. Problemy tego rodzaju J. Kozielecki nazywa problemami otwartymi. Rozwiązanie ich wymaga istnienia w programie zarówno generatora, jak i ewaluatora pomysłów.



W wielu przypadkach rozwiązanie problemu wymaga jedynie zastosowania jednego ze znanych sposobów. Problemy tego typu zwane są zamkniętymi. Programy rozwiązujące problemy zamknięte nie muszą w swej strukturze mieć generatorów pomysłów, a jedynie ewaluatory, które służą do wyboru właściwej metody ze zbioru metod z góry określonych.

Programowanie heurystyczne jako metoda rozwiązywania trudnych problemów za pomocą maszyn elektronicznych nie zostało jeszcze zbadane dostatecznie dokładnie. Przeprowadzone dotychczas studia pozwoliły zaledwie stwierdzić przydatność tej metody oraz to, że pełniejsze jej poznanie wymaga przeanalizowania wielu interesujących zagadnień. Do tych ostatnich należą bez wątpienia zagadnienia wytwarzania skutecznych heurystyk, tzn. metod, którymi posługuje się człowiek w swojej codziennej pracy. Badania nad zagadnieniami dotyczącymi procesów myślenia pozwalają stwierdzić, że reguły heurystyczne człowiek przyswaja sobie w toku uczenia się lub w czasie rozwiązywania problemów. Stwierdzono przy tym, że niezmiernie istotną rolę odgrywa tu wyrobienie umiejętności dostrzegania takich własności rozpatrywanych obiektów, które pozwalają szybciej znaleźć rozwiązanie.

Zmusza to badaczy zagadnień sztucznej inteligencji, a więc i badaczy zagadnień programowania heurystycznego do poszukiwania dróg „nauczania" komputera. Zagadnienie to wysuwane jest przez wielu autorów jako jedno z najważniejszych. Uważają oni, że w istocie heurystycznymi są tylko te programy, które potrafią doskonalić zaprogramowane w nich metody rozwiązywania problemów.

Obecnie trudno jest przewidzieć, w jakim kierunku rozwiną się rozmaite gałęzie heurystyki, jako nauki. Pewne jest jednak, że bez rozwoju badań nad sztuczną inteligencją, a w tym — nad programowaniem heurystycznym — postęp w dziedzinie twórczości intelektualnej będzie utrudniony. Przezwyciężenie tych trudności wiąże się z poznaniem struktur i mechanizmów procesów myślowych tj. z poznaniem heurezy. Otworzy to drogę do szerszego wykorzystania komputerów. B. Puszkin, rozważając te kwestie, stwierdza: „Myśl twórcza uczonych i techników to największe bogactwo ludzkości. Bogactwo to należy wykorzystywać nie mniej intensywnie i celowo niż np. skarby wydobywane z głębi ziemi" (B. Puszkin 1967, wyd. polskie 1970, str. 238).

LITERATURA

[1] Apter M. J.: Komputery a psychika — symulacja zachowania; Hutchinson a. Co. Pub. Ltd., Londyn 1970; wyd. polskie — PWN, Warszawa 1968

[2] Feigenbaum E. A., Feldman J.: Maszyny matematyczne i myślenie; McGraw — Hill Book Comp., New York, 1963; wyd. polskie — PWN, Warszawa, 1972

[3] Gelernter H.: Realization of a geometry theorem-proving machine; w zbiorze: Proc. of the Int. Conf. on Information Processing, Paris, UNESCO House, 1959

[4] Kozielecki J.: Zagadnienia psychologii myślenia; PWN, Warszawa, 1968

[5] Ledley R. S.: Programming and utilizing digital computers; McGraw — Hill Book Comp., N. York, 1962

[6] Newell A., Shaw J. C., Simon H. A.: Empirical explorations with the logic theory of machine; ref. West. Joint Comp. Conf. 1957

[7] Newell A., Shaw J. C., Simon H. A.: A general problem — solving program for a computer; „Computers and Automation" 8/1959

[8] Puszkin B.: Ewristika — nauka o tworczieskom myszlenii; wyd. „Izdatielstwo Politczieskoj Litieratury", Moskwa, 1967; wyd. polskie — „Książka i Wiedza", Warszawa, 1970

[9] Slagle J.: Artificial intelligence: the heuristic programming approach; McGraw — Hill Book Comp., N. York, 1971

[10] Wyman F. P.: Binary programming: a decision rule for selecting optimal of heuristic techniques; „The Computer Journal", vol. 16, 2/1973

Jurg NIEVERGELT
Wydział Informatyki Uniwersytetu w Illinois
USA

681.322.004.14:371.3:378.4:51

ACSES — zautomatyzowany system nauczania informatyki

Przedstawiono projekt systemu dla uniwersytetów amerykańskich, opierający się na systemie PLATO IV.

W odpowiedzi na olbrzymi (obecnie 2000 studentów na semestr) i wzrastający napływ studentów na wstępne kursy z informatyki, rozpoczęliśmy realizację zamierzenia, które ma na celu zautomatyzować te kursy. Opracowaliśmy podsystem nauczania informatyki, oparty o system skomputeryzowanego nauczania PLATO IV na Uniwersytecie Illinois. W zamierzeniach, kurs ten jest przedmiotem dodatkowym na naszym Uniwersytecie, jednocześnie jest on zaprojektowany w ten sposób, ażeby mógł być użyteczny jako całkowicie niezależny system do wstępnego nauczania informatyki, nawet w odległych ośrodkach.

Najistotniejsze składniki tego systemu to:

- biblioteka wykładów, obejmująca kilka języków programowania, metod obliczeniowych i dziedzin zastosowań
- całkowicie niezależny system programowania konwersacyjnego: przygotowywanie, wykonywanie i uruchamianie programów napisanych przez studentów w jednym z języków objętych wykładami
- programy oceniające programy napisane przez studentów, uwzględniające poprawność i elegancję wykonania
- konwersacyjny doradczy i informacyjny system wyszukiwawczy jako przewodnik dla studenta w bibliotece wykładów, biorący zamierzenia studenta i jego dotychczasowe wyniki
- system porozumiewania, który pozwala studentowi kontaktować się z wykładowcą-człowiekiem.

W artykule tym opisane są założenia i obecny stan projektu. Komputery odgrywają coraz istotniejszą rolę w naszym społeczeństwie, co prowadzi bezpośrednio do gwałtownie wzrastającego zapotrzebowania na podstawowe wykształcenie informatyczne. Popyt ten wypływa z dwóch źródeł:

Po pierwsze, ciągle wzrasta zapotrzebowanie na specjalistów informatyków komputerowych. Powszechnie uważa się, że w ciągu 5 lat nastąpi podwojenie całkowitego zapotrzebowania na analityków systemów, programistów, operatorów komputerów i konserwatorów.

Po drugie, można oczekiwać, że większość ludzi będzie musiała współdziałać z komputerami w swojej codziennej pracy, i to już w najbliższych 10, 20 latach. Nawet ludzie, nie związani bezpośrednio z komputerami, powinni mieć pewne pojęcie o nich, ponieważ, jeśli chcemy podejmować prawidłowe decyzje dotyczące przyszłej roli komputerów w naszym społeczeństwie, musimy oprzeć się o ludzi z odpowiednim przygotowaniem. Obecnie, wielu obywateli patrzy na komputery z obojętnością, niektórzy obawiają się maszyn, widząc w nich zagrożenie swoich praw, bezpieczeństwa i godności. Taki stosunek oczywiście nie wystarczy. Stąd, ważnym jest, aby każdy wykształcony człowiek rozumiał podstawowe zasady działania komputerów i ich znaczenie dla społeczeństwa, jak również posiadał pewne umiejętności w ich wykorzystywaniu.

I w końcu każdy powinien mieć możliwość przyswajania sobie wiedzy w najdogodniejszy dla niego sposób. Biorąc pod uwagę doniosłość „elementarza komputerowego". Doradczy Komitet Naukowy Prezydenta zalecił w swoim raporcie z 1967 r., ażeby 75% wszystkich studentów miało dostęp do komputerów. Chociaż dotychczas procent ten nie został osiągnięty, zapotrzebowanie na nauczanie podstaw informatyki na naszych uniwersytetach, w szkołach wyższych, w szkołach pomaturalnych i prywatnych szkołach elektronicznego przetwarzania danych, jest olbrzymie. Chociaż nie posiadamy obecnie dokładnych danych, możemy dostrzec ogólną tendencję, przeglądając dane z lat 65—69 zestawione w *Computer Sciences Project*, i przygotowane przez Komitet Oświaty Regionu Południowego [1]. W latach 1964—65 było około 4300 studentów niższych lat i 1300 — wyższych, których głównymi przedmiotami były: informatyka, przetwarzanie danych, teoria informacji itd. W latach 1966—1967 liczby te podskoczyły do 22000 i 5000 odpowiednio. Dodatkowo jeszcze, przy zajęciach związanych z pracami dyplomowymi, 120000 studentów niższych lat i 29000 — wyższych w latach 1964—1965 otrzymało pewne przeszkolenie komputerowe. Do roku 1968—1969 liczby te wzrosły niemal trzykrotnie — do 350000 i 81000 odpowiednio. Dane wskazują również, że większość tych studentów, wyposażenia i dotacji było skoncentrowane w stosunkowo niewielu uniwersytetach i szkołach wyższych. Lecz mniejsze szkoły wyższe są również zainteresowane w realizacji programu studiów w zakresie informatyki. Amerykański Związek Szkół Pomaturalnych w swoim raporcie *„The Computer and the Junior College"* [2] stwierdził, że „środowisko uczelni musi przyjąć wyzwanie i pomóc wypełnić zapotrzebowanie... dla praktyków z doświadczeniem (...poprzez solidne programy wiedzy technicznej w zakresie przetwarzania danych w przedsiębiorstwie)". Niestety problemy natury finansowej łączące się z takimi programami, wstrzymały wiele szkół od zajęcia się tą dyscypliną. Instytucje te oparły się na tradycyjnym podejściu dydaktycznym wykład-dyskusja-laboratorium. Metoda ta ma wiele braków, szczególnie wtedy, kiedy ma się do czynienia z dużą liczbą studentów. Nie odpowiada to zupełnie charakterowi przedmiotu i powoduje zniechęcenie u studentów. Uczenie się programowania wymaga aktywnego uczestnictwa i dużego wysiłku ze strony studenta — dwóch rzeczy, których nie daje metoda nauczania drogą wykładu. Po jednym wykładowcy dla każdego studenta, byłoby sytuacją idealną przy nauce umiejętności, takiej jaką jest programowanie, ale ten rodzaj instruktażu jest oczywiście ekonomicznie niemożliwy do spełnienia wtedy, kiedy stawia się na edukację masową.

[1]) „Computers in Higher Education", Southern Regional Education Board, Atlanta, Georgia, August 1967.
[2]) „The Computer and the Junior College", American Association of Junior Colleges, Washington, D.C., 1969.

W dodatku, nauczanie metodą tradycyjnych wykładów może objąć ograniczone audytorium. W szczególności wyłącza to wszystkich ludzi, których obowiązki zawodowe nie pozwalają na uczęszczanie do szkoły.

Należy oczekiwać dużego zapotrzebowania na podstawowe kursy informatyki w związku ze stałym dokształcaniem dorosłych. Sytuacja, w której ktoś nagle stwierdzi, że powinien wiedzieć coś na temat komputerów, ażeby móc nadal pracować efektywnie, będzie stawała się coraz bardziej powszechna.

Wszystkie te wybiegające w przód rozważania, w połączeniu z naszym doświadczeniem w nauczaniu wstępu do informatyki na Uniwersytecie Illinois (około 2000 studentów na semestr, z różnych środowisk), zmusił nas do rozpoczęcia prac nad realizacją kompleksowego projektu zautomatyzowania wstępnych kursów z informatyki.

System PLATO IV, opracowany przez Laboratorium Badawcze Nauczania Komputerowego na Uniwersytecie Illinois daje nam wyjątkowe możliwości przeprowadzenia zautomatyzowanego kursu składającego się wykładów prowadzonych metodą CAI *(Computer Aided Instruction)**) o komputerach i ich oprogramowaniu oraz wypróbowania tego systemu na dużym audytorium o różnym wykształceniu. Obecnie system PLATO IV działa i w najbliższej przyszłości ma mieć ponad 1000 urządzeń końcowych, umiejscowionych w szkołach i uczelniach z różnego rodzaju społecznością studencką.

Potencjalnie, zautomatyzowane nauczanie metodą CAI ma wiele zalet: zapewnia prawdziwie indywidualizowany instruktaż pozwalający studentom zapoznanie się z tokiem wykładów, przykrojonych do ich potrzeb i w ich tempie, a mając odpowiednią sieć urządzeń końcowych może objąć szerokie audytorium i w najbliższej przyszłości — małym kosztem. Potencjalna efektywność metody CAI wskazuje że dla szkół i uczelni, które nie organizują jeszcze kursów z informatyki, może stać się najtańszym sposobem wprowadzenia tego rodzaju programów studiów.

Jednakże te zalety metody CAI zostaną zrealizowane dopiero wtedy, kiedy zwiększy się wysiłki w zakresie prac badawczo-rozwojowych. Stawiamy nasz projekt jako wkład w dążeniu do wykazania możliwości podejścia zautomatyzowanego nauczania do problemu masowej edukacji podstaw informatyki.

ACSES, nasz zautomatyzowany system nauczania informatyki opracowany w oparciu o system PLATO IV, jest zaprojektowany tak, aby mógł służyć w dwojaki sposób, zgodnie z dwoma celami: jako szkolenie dodatkowe na naszym uniwersytecie i podstawowe szkolenie w odległych ośrodkach.

Częściowo zautomatyzowany sposób szkolenia dodatkowego

Na naszym uniwersytecie, odpowiedzialnym za kurs w dalszym ciągu będzie wykładowca a wykłady metodą CAI mają służyć jako uzupełnienie.

Wykładowca przedyskutuje problem w klasie i odeśle studentów do odpowiedniego wykładu, który da więcej szczegółów i przykładów, i umożliwi studentowi ćwiczenie lub rozwiązanie problemu na komputerze. Ten częściowo zautomatyzowany sposób jest łatwy lub trudny do osiągnięcia przy współczesnych środkach technicznych, zależnie od tego, jak skomplikowane mają być te wykłady. Bezpośrednie zajęcia lub ćwiczenia nie są trudne do przygotowania, lecz wykład, który potrafi ocenić złożoną odpowiedź studenta może być bardzo trudny do realizacji.

W przypadku naszych wykładów z informatyki, odpowiedź do oceny jest często programem, przygotowanym przez studenta.

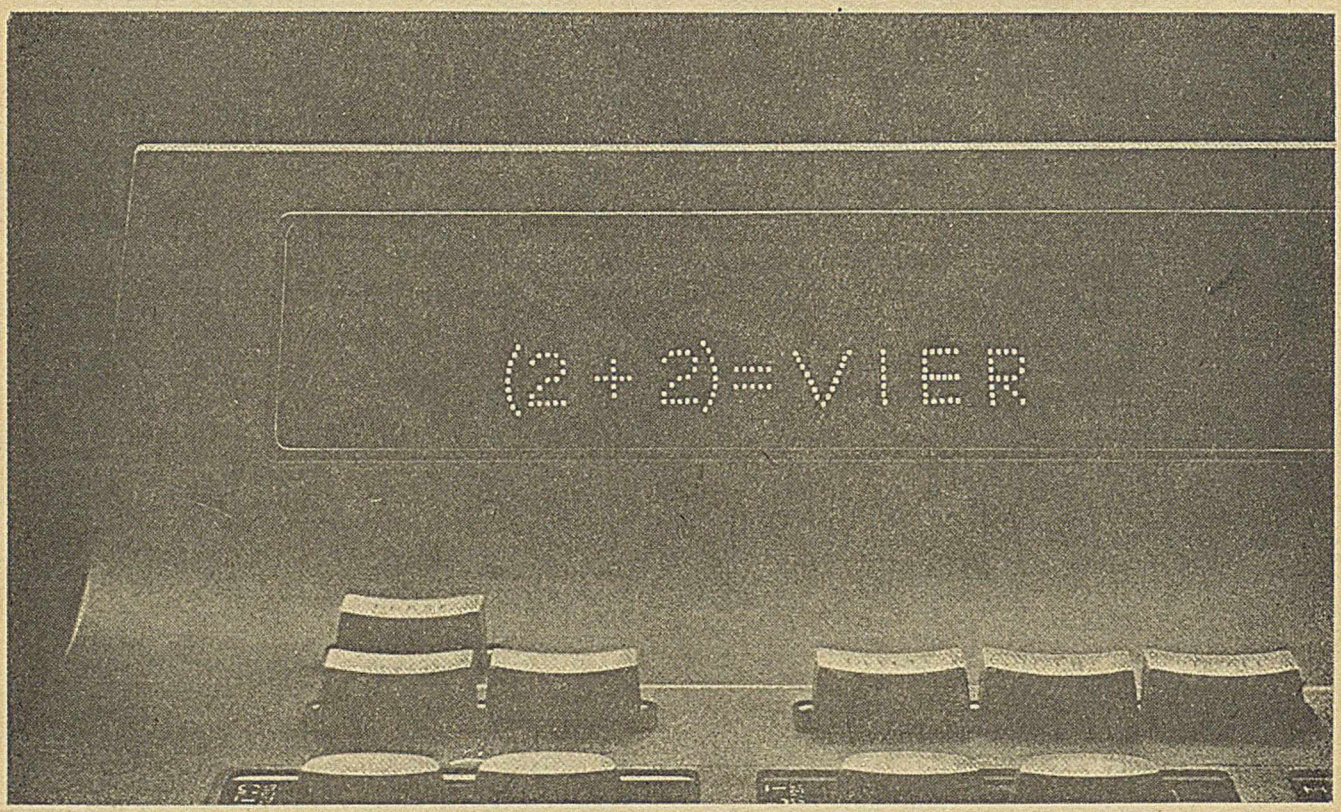
Chcielibyśmy ocenić go pod względem poprawności jak i elegancji opracowania.

A zatem, w celu realizacji częściowo zautomatyzowanego kursu z informatyki potrzebujemy przede wszystkim:

— bibliotekę wykładów obejmującą kilka języków programowania, technik obliczeniowych i dziedzin zastosowań

---

*) nauczanie wspierane komputerem.



— programy oceniające programy napisane przez studentów, które są odpowiedzią na dobrze postawione problemy.

Całkowicie zautomatyzowany sposób szkolenia podstawowego.

Spodziewamy się, że zapotrzebowanie na podstawowe kursy z informatyki ze strony szkół średnich i wyższych oraz kursów stałego dokształcania dorosłych wzrośnie gwałtownie w najbliższej przyszłości. W tych środowiskach może nie być wykładowcy, który poprowadzi studenta i wypełni braki, które mogą znajdować się w materiale wykładu. Niemniej jednak w tym środowisku system powinien być użytkowany w postaci całkowicie zauotmatyzowanej. Proponujemy zrealizować przez dostarczenie:

- całkowicie niezależnego systemu programowania konwersacyjnego do przygotowania, wykonywania i uruchamiania programów napisanych przez studenta w jednym z języków objętych wykładami
- konwersacyjnego doradczego i informacyjnego systemu wyszukiwawczego, jako przewodnika dla studenta w bibliotece wykładów, biorąc pod uwagę zamierzenia studenta i jego dotychczasowe wyniki
- system porozumienia, który pozwala studentowi kontaktować się z wykładowcą-człowiekiem, w celu otrzymania pomocy i rady, z jakiejkolwiek końcówki podłączonej do systemu PLATO.

Proponujemy wykorzystać możliwości jakie posiadamy, ażeby zautomatyzować wstępne kursy z informatyki na naszym uniwersytecie, traktując to jako główny wysiłek badawczy, pozwalający stwierdzić, jak dalece kursy wstępne można zautomatyzować. Jesteśmy przekonani, że nadszedł czas, ażeby pchnąć technologię szkolenia wspieranego przez komputer na poziom, gdzie może ona zastąpić pełny kurs lub zestaw kursów zbliżonych. Sądzimy również, że jeśli niezbędne pomysły i wysiłek zostaną włożone w zaprojektowanie zautomatyzowanego kursu i oprogramowania, które czyni go możliwym, kurs taki może przynieść lepsze wyniki nauczania, od tych, które student osiąga na rozbudowanych kursach wstępnych prowadzonych w oparciu o metodę konwencjonalnych wykładów.

Pozostała część artykułu opisuje bardziej szczegółowo części składowego zautomatyzowanego kursu, który aktualnie rozwijamy, jego obecny zaawansowany stan i procedury które zamierzamy wykorzystać, ażeby oceniać poszczególne wykłady i cały kurs.

Biblioteka wykładów

Wstępne kursy z informatyki przygotowywane są dla studentów z różnych środowisk o różnych zainteresowaniach, dlatego też kursy różnią się od siebie znacznie, np. zagadnienia poruszane na kursach komputerowych dla inżynierów, różnią się od kursów dla studentów ekonomistów lub nauczycieli. Co więcej, wzrastająca liczba studentów zapoznaje się z obliczeniami w szkole średniej, co znaczy, że musimy brać pod uwagę ludzi o pewnym stopniu znajomości komputera, jak również osób całkowicie niewtajemniczonych. Z drugiej strony, kursy te pokrywają się, w momencie, kiedy dotyczą podstawowych struktur właściwych językom programowania wyższego rzędu. Z tego powodu zdecydowaliśmy się nie układać wykładów w cykle, dla każdego z wielu kursów osobno[3]), zamiast tego, tworzymy pokaźną bibliotekę wykładów, skąd każdy wykładowca wstępnego kursu z informatyki może wybrać odpowiedni zestaw.

Zrobiono wszystko, ażeby zapobiec „kartowaniu" lekcji, w których co krok mamy pytania o wielu możliwych odpowiedziach. Końcówka PLATO IV jest siatką 512 na 512 indywidualnie sterowanych punktów i jako taka może z łatwością pokazywać dwuwymiarowy materiał, o pewnym, chociaż powolnym ruchu.

Założeniem naszego projektu wykładów jest wykorzystanie tego urządzenia wejścia/wyjścia w celu ożywienia przedstawionego materiału.

Każdy z około stu wykładów naszego systemu dostarcza prawie godzinnego instruktażu. Wykłady te dzielą się ogólnie na trzy kategorie: wykłady wstępne, które nie wymagają żadnego przygotowania, wykłady, które omawiają gramtykę i semantykę różnych języków programowania i wykłady omawiające zastosowania komputerów. Uważa się, że wśród tych wykładów z trzech różnych dziedzin, właściwie każdy wykładowca znajdzie duży, przydatny zestaw.

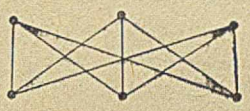
Wykłady wstępne zwykle zmierzają do przedstawienia kilku podstawowych pojęć w sposób uproszczony. Na przykład wykład nazwany Algolingo, zapoznaje studenta z pojęciem algorytmu. W Algolingo, przedstawiona jest metoda znajdowania w całym zbiorze wyników najlepszego wyniku i taki „bezpośrednio wiodący do celu" algorytm jest opisany w różnych „językach". Najpierw algorytm ten jest omawiany szczegółowo przy czym ukazuje się opis prozy angielskiej, po nim następuje algorytm rymowany, na koniec, algorytm jest ukazany w postaci prostych programów w FORTRAN, PL/1 i BASIC, przy czym każde zdanie jest szczegółowo komentowane, dla wyjaśnienia jego celu.

Często, „podstawowe pojęcie" jest przedstawione za pomocą „mini języka programowania"[4]), który został starannie pomyślany, specjalnie po to, by uwydatnić to pojęcie, bez wchodzenia w inne tematy. Typowym przykładem jednego z takich języków jest DOODLE *(Do-loops Only Drawing Language)*, język wprowadzający pojęcie pętli. DOODLE ma tylko cztery typy zdań: pętle do-end, zdania podstawień arytmetycznych, zdania rysowania linii i zdania wymazywania linii.

Na przykład program

```
do a 1, 3
do b 1, 3
draw 40 + 40a, 100; 40 + 40b, 500;
end
end
```

powoduje narysowane figury (rys. 1).



Studenci piszą i redagują programy w DOODLE konwersacyjnie i mogą je wykonać, przy czym wyniki pokazują się na ekranie. Pomimo swojej prostej struktury, DOODLE może być użyty do uzyskiwania zupełnie skomplikowanych rysunków; np. obrazy graficzne dobrze znanej *„curve stifchings"* mogą być z łatwością zaprogramowane w DOODLE.

Typowym cyklem wykładów dla różnych języków programowania jest cykl wykładów PL/1, który obecnie składa się z około tuzina wykładów z naświetleniem różnych aspektów języka. Są wykłady przedstawiające typy danych języka PL/1, operacje na tych typach danych, zdania IF, zdania DO, wejście/wyjście i procedury. Na końcu jest wykład, który jest testem sprawdzającym zaawansowanie studenta w opanowaniu PL/1. Inne cykle wykładów rozpoczęto dla języków BASIC, FORTRAN i COBOL.

Wykłady na temat różnych zastosowań są bardzo urozmaicone, obejmują takie problemy jak sortowanie, metody Monte Carlo, symulację, rozwiązywanie równań liniowych z wieloma niewiadomymi i rozwiązywanie nieliniowych równań z jedną niewiadomą.

[3]) Fenichel, R. R., Weizenbaum and J. C. Yochelson, „A program to teach programming", **Comm. ACM** 13 (1970), 141—146.

[4]) Ledgard, H. F., „Ten mini-languages: A study of Topical Issues in Programming Languages", COMPUTING SURVEYS 3 (1971), 115—146.

Wykłady dotyczące sortowania opisują różne metody sortowania, takie jak sortowanie przez przesuwanie, sortowanie przez wstawianie i szybkie sortowanie *(quicksort)*. Studenci mają podać ciągi liczb do sortowania oraz samodzielnie posortować zadane ciągi.

Cykl dotyczący analizy numerycznej składa się obecnie z wykładów na temat rachunku najmniejszych kwadratów, całkowania numerycznego, rozwiązywania równania nieliniowego, rozwązywania układu równań liniowych i programowania liniowego. PLATO IV idealnie odpowiada nauczaniu analizy numerycznej, ponieważ jego graficzne i konwersacyjne możliwości obliczeń jasno ukazują podstawowe pojęcia tej analizy. Dla przykładu, pojęcia będące podstawą metody znajdowania pierwiastka, stają się przejrzyste, kiedy wyrażone są geometrycznie i przedstawione graficznie, a pojęcie iteracji staje się zrozumiałe dla studenta obserwującego działanie typowych schematów znajdowania pierwiastka, zastosowanych do funkcji przez niego wybranych. Dalej, możliwości graficzne dają bodźce do rozwoju algorytmów numerycznych, np. w wykładzie na temat najmniejszych kwadratów, student najpierw dokonuje eksperymentu mierząc powiększenie się „rubber-band", spowodowane różnymi wagami. Następnie, mając do czynienia z problemem dopasowania prostej linii do swoich danych, wprowadza się studenta w aproksymację metodą najmniejszych kwadratów.

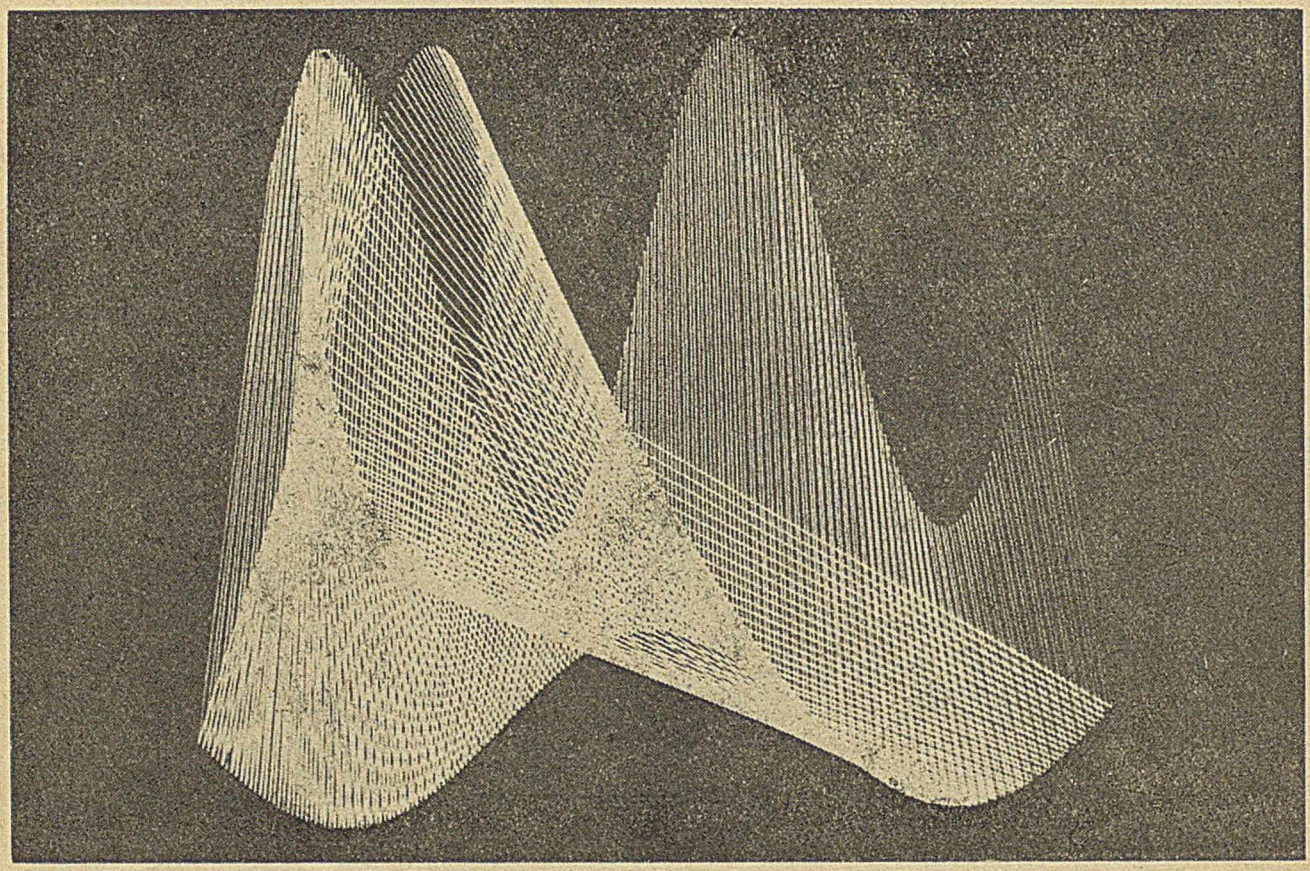
Wykład na temat metod Monte Carlo omawia w jaki sposób przy pomocy przypadkowości można otrzymać rozwiązania numeryczne deterministycznych problemów takich, jak: znajdowanie powierzchni pod krzywymi, rozwiązywanie równań potencjału. Student rysuje jakikolwiek wielokąt a następnie obserwuje jak obliczone jest jego pole przez wybór przypadkowych punktów i wyliczanie ułamka tego pola znajdującego się wewnątrz wielokąta. Podobnie, student może określić jakiś obszar wielokąta oraz warunki brzegowe, a następnie obserwować jak poruszanie się w przypadkowych kierunkach prowadzi do aproksymacji temperatury w jakimś punkcie wewnętrznym, wybranym przez studenta.

Myślą przewodnią innych wykładów dostępnych dla studenta jest przekonanie go, że programowanie komputerowe może być zarówno przyjemne jak i prowokujące. Np. w wykładzie zwanym SOMAGA *(Software Management Game)*, współzawodniczą ze sobą w próbach napisania programu w DOODLE w celu narysowania bardziej skomplikowanej figury.

Studenci ustalają sobie czas do wykonania zadania, a następnie muszą napisać program tak, by się zmieścić w ustalonym czasie. W innych wykładach, które są aktualnie opracowywane, studenci piszą programy gry nazwanej Darwin[5], w której programy te próbują się same reprodukować i zniszczyć programy przeciwne. W czasie tych wykładów istotny jest fakt, że końcówki PLATO mogą się wzajemnie komunikować.

---

[5] X. (pseudonim) Computer Recreations: Darwin, SOFTWARE-PRACTICE AND EXPERIENCE 2 (1972), 93—96.



Lesław Aleksander SIENIAWSKI
Instytut Cybernetyki Technicznej Politechniki Wrocławskiej

681.322.004.14:65.011.56

O pojęciu i celu identyfikacji informacyjnej

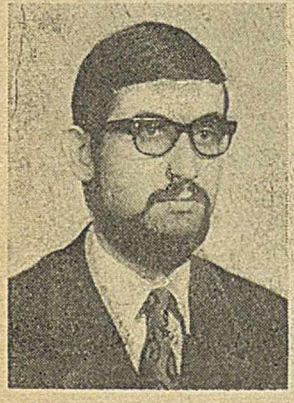
Podjęto próbę sformułowania podejścia do rozpoznawania właściwości procesów informacyjnych dla potrzeb projektowania systemów informatycznych i oceny doboru środków.

Wstęp

Od momentu powstania von neumannowskiej koncepcji maszyny cyfrowej (EDVAC, rok 1945) minęło blisko ćwierć wieku. W ciągu tego okresu powstawały kolejne generacje maszyn wykorzystujące najnowsze zdobycze technologii i charakteryzujące się coraz bardziej rozbudowaną architekturą. Ewolucji podlegała również świadomość użytkowników. Do przeszłości należy już okres, w którym użytkownikowi komputera wystarczał sam fakt uzyskiwania poprawnych wyników obliczeń. Od sprzętu liczącego i jego oprogramowania żąda się obecnie spełnienia określonych postulatów dotyczących z jednej strony łatwości programowania i bezpośredniego kontaktu użytkownika z maszyną, automatyzacji czynności operatorskich i administracyjnych, odpowiedniej szybkości i niezawodności, zaś z drugiej — obniżenia wielkości nakładów inwestycyjnych i eksploatacyjnych. Pogodzenie wynikających z tego sprzecznych tendencji nie jest jednak łatwe, co wypływa m. in. z braku aparatu formalnego dostatecznie wiernie i wszechstronnie opisującego skomplikowane zjawiska zachodzące w komputerze. Dodatkową trudność sprawia określenie ilościowych charakterystyk wymagań nakładanych na sprzęt i oprogramowanie. Ranga zagadnienia rośnie ze wzrostem objętości przetwarzanych zbiorów informacji, złożonością struktury procesu przetwarzania informacji oraz z układową i programową rozbudową środków przetwarzania.

Autorzy raportu [3] stwierdzają brak apriorycznych metod oceny jakości doboru środków przetwarzania do konkretnych zastosowań oraz pilną potrzebę opracowania takich metod. Podobnych wniosków dostarcza przegląd literatury z tego zakresu. Rozważane są zazwyczaj problemy cząstkowe, niekiedy bez określonych ograniczeń [2] oraz bez metodologicznego umiejscowienia ich w ogólnym problemie syntezy systemów informatycznych.

W artykule niniejszym podjęto próbę wypełnienia metodologicznej luki w tym zakresie.



Mgr inż. LESŁAW ALEKSANDER SIENIAWSKI (1949) jest absolwentem Wydziału Elektroniki Politechniki Wrocławskiej, który ukończył (1972) ze specjalnością maszyny matematyczne. Obecnie jest doktorantem w Instytucie Cybernetyki Technicznej i zajmuje się problematyką badania własności procesów informacyjnych.

Definicje

W punkcie tym podamy bliższe określenia wykorzystywanych dalej pojęć.

Przez **algorytm przetwarzania informacji** (algorytm) będziemy rozumieli dowolny sformalizowany opis czynności jakie należy wykonać na wartościach danych, zwanych danymi wejściowymi.

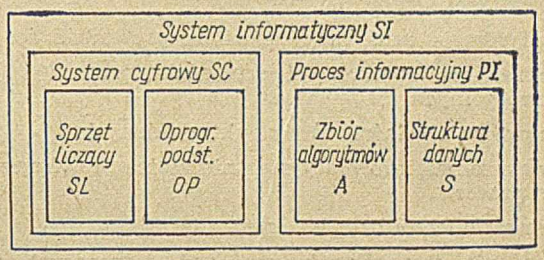
Przez strukturę danych **związaną** ze zbiorem algorytmów A będziemy rozumieli taką strukturę danych S, że dla każdego algorytmu $a \in A$, jego dane wejściowe i wyjściowe należą do tej struktury danych.

Procesem informacyjnym (PI) nazwiemy obiekt złożony ze zbioru algorytmów A i struktury danych S związanej z tym zbiorem.

Systemem cyfrowym (SC) nazwiemy obiekt złożony z dwu obiektów: sprzętu liczącego (SL) i oprogramowania podstawowego (OP).

Z kolei **systemem informatycznym** (SI) będziemy nazywać obiekt złożony z procesu informacyjnego i systemu cyfrowego. Zestaw środków sprzętowych i programowych tworzących system cyfrowy jest przeznaczony do realizacji zadań określonych przez proces informacyjny.

Ilustracją powyższych definicji jest rys. 1.



Rys. 1. System informatyczny jako obiekt złożony

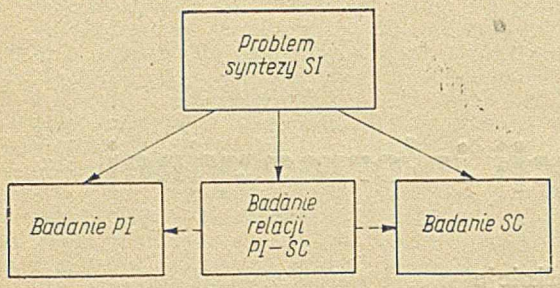
Pojęcie identyfikacji informacyjnej

Można wyróżnić dwa biegunowe przypadki znajomości procesu informacyjnego:

- brak informacji
- pełna znajomość — znana definicja procesu oraz znane zbiory wartości danych wejściowych.

Przypadek pierwszy wyklucza analizę procesu, przypadek drugi jest reprezentatywny dla obliczeń naukowo-technicznych. Sytuacje rzeczywiste mieszczą się „pomiędzy" wymienionymi z włączeniem przypadku drugiego. Poszczególnym stopniom znajomości procesu przypisuje się określoną metodykę syntezy systemu informatycznego, tj. doboru systemu cyfrowego do realizacji zadanego procesu, z uwzględnieniem przyjętych ograniczeń. W zakres syntezy wchodzi dobór sprzętu liczącego i jego oprogramowania podstawowego oraz określenie sposobu współpracy poszczególnych elementów (por. rys. 1). Podkreślić należy, że — ze względów o których była mowa wcześniej — dobór ten jest zazwyczaj dokonywany w oparciu o analizę mniej lub bardziej prymitywnego modelu rzeczywistości, wspieraną doświadczeniem i intuicją projektanta.

Zagadnienie syntezy systemu informatycznego może być sprowadzone do trzech węższych i prostszych lecz wzajemnie powiązanych problemów (rys. 2.):



Rys. 2. Dekompozycja problemu syntezy systemu informatycznego

- badania właściwości procesów informacyjnych
- badania charakterystyk systemów cyfrowych
- określenia wzajemnych zależności między procesami informacyjnymi a systemami cyfrowymi.

W chwili obecnej najbardziej rozwinięte są tu metody oceny właściwości systemów cyfrowych (komparatystyka, mieszanki obliczeniowe, zadania porównawcze, monitorowanie) [2]; nie znaczy to bynajmniej, że stan wiedzy w tym zakresie jest zadowalający. Poszukiwanie wszechstronnej charakterystyki danego systemu cyfrowego wymaga uprzedniego zdefiniowania samego pojęcia takiej charakterystyki, co nie jest rzeczą łatwą, zważywszy znaczną różnorodność istniejących rozwiązań sprzętowo-programowych. Ogólne pojęcie rozpoznania właściwości procesów informacyjnych, z punktu widzenia syntezy systemu informatycznego — tzw. **identyfikacji informacyjnej** — pochodzi od J. Bromirskiego. Celem rozpoznania prowadzonego niezależnie od sposobu technicznej realizacji procesu jest zdobycie możliwie najlepszej charakterystyki procesu, tj. określenie cech, od których zależeć może wynik syntezy SI. Szczegółowe cele identyfikacji omówimy poniżej.

Identyfikacja informacyjna winna prowadzić do:

- określenia struktury i semantyki procesu i jego poszczególnych fragmentów
- określenia właściwości struktury danych
- wskazania informacyjnie niezależnych fragmentów procesu
- wskazania możliwości optymalizacji
- podania szczególnych wymagań dotyczących projektowanego systemu cyfrowego.

Określenie semantyki algorytmów wchodzących w skład procesu umożliwia sprawdzenie ich merytorycznej poprawności, tj. zgodności realizowanych przez nie odwzorowań wartości danych z przekształceniami założonymi. Oprócz tego znajomość semantyki stanowi podstawę porównywania algorytmów równoważnych pod względem złożoności i poszukiwania algorytmów optymalnych. Nie bez znaczenia są też możliwości uproszczeń nie zachowujących semantyki (dokonywane w porozumieniu z przyszłym użytkownikiem systemu).

Badanie właściwości struktury danych jest nieodzowne z punktu widzenia projektowania pojemności, typu i organizacji pamięci systemu cyfrowego. Jednym z zasadniczych parametrów jest tu objętość informacyjna struktury danych, związana z formatami przedstawienia poszczególnych danych w pamięci.

Określenie niezależnych fragmentów procesu, jako wynik analizy jego struktury, stanowić ma podstawę do rozważań dotyczących organizacji przetwarzania, tj. rozdziału zasobów systemu, a w szczególności: czasu, procesorów i pamięci. Celowi temu służy również wyznaczanie czasowych charatkerystyk algorytmów.

Pod pojęciem „wymagania szczególne" rozumiemy wymagania wynikające bezpośrednio z natury zjawisk obsługiwanych przez dany proces informacyjny, a między innymi: punkty nadawania, odbioru, magazynowania i przetwarzania informacji (struktura przestrzenna SI), rodzaj urządzeń wejścia/wyjścia, urządzeń transmisyjnych, zakres i dokładność przedstawiania liczb, itp.

W konkretnej sytuacji zadanie identyfikacji może być ograniczone do wybranych cech procesu. Może to wynikać np.:

- z małej znajomości procesu
- z narzucenia na projektowany system ograniczeń łatwych do spełnienia
- z dysponowania niewielkim repertuarem sprzętu liczącego i oprogramowania podstawowego
- z przesłanek ekonomicznych lub organizacyjnych.

Stosownie do konkretnie postawionego celu, problem identyfikacji można rozwiązywać w różnych skalach, tj. z różnym stopniem szczegółowości:

- w skali **mikro** — na poziomie elementarnych operacji przetwarzania (np. dodawanie, skok)
- w skali **makro** — na poziomie wyodrębnionych fragmentów procesu, tzn. układów operacji elementarnych (np. podprogramy, programy).

Efekt identyfikacji zależy oczywiście od przyjętej skali. Intuicyjnie jest jasne, że wraz ze wzrostem szczegółowości otrzymywane wyniki coraz lepiej opisują proces. Nie znaczy to jednak, że w każdej konkretnej sytuacji należy dążyć do uzyskania maksymalnej szczegółowości.

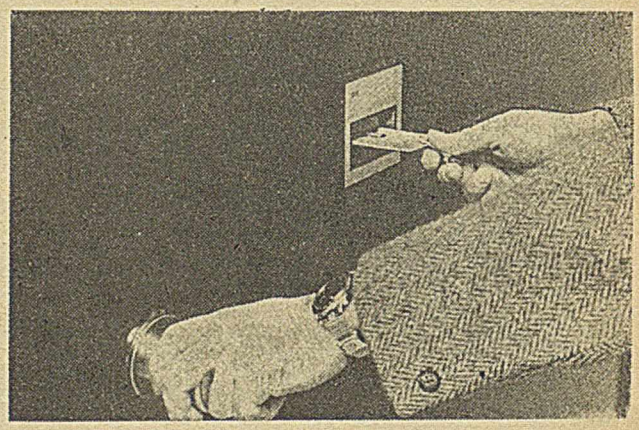
Język opisu

Warunkiem koniecznym identyfikowania procesu informacyjnego jest posiadanie jego definicji — opisu w pewnym mniej lub bardziej sformalizowanym języku. Wydaje się, że język taki powinien:

- być niezależny od systemu cyfrowego, który ewentualnie mógłby być użyty do realizacji danego procesu
- dać możliwość przedstawienia procesu z dowolną szczegółowością
- mieć prostą interpretację graficzną
- bazować na jednorodnym aparacie matematycznym.

Wśród istniejących języków (metod) opisu procesów informacyjnych wyróżnimy następujące klasy:

- sieci działań
- metody grafologiczne
- schematy logiczne (Janowa)
- metody algebraiczne
- języki programowania.



Dokonamy teraz krótkiego przeglądu możliwości wykorzystania poszczególnych klas.

Język **sieci działań** jest podstawowym narzędziem definiowania algorytmów wchodzących w skład procesu. Wyrażenia tego języka są przejrzyste, dają pogląd o czynnościowej strukturze procesu. Umożliwiają opis zarówno operacji elementarnych jak i układów takich operacji. Sieć działań może być, w ograniczonym zakresie, pomocna przy optymalizacji procesu.

Metody grafologiczne różnią się między sobą interpretacją krawędzi i wierzchołków oraz dopuszczeniem (lub nie) cykli. Niekiedy rozróżnia się typy wierzchołków. Istniejące metody tej klasy dają możliwość przedstawienia struktury czynnościowej z dowolną szczegółowością oraz optymalizacji ze względu na czas przetwarzania. Nie opisują semantyki.

Schematy logiczne procesów umożliwiają przedstawianie ich struktury oraz badanie równoważności; stopień szczegółowości w zasadzie dowolny. Nie opisują semantyki i struktury danych.

Metody algebraiczne są najbardziej rozwinięte. Wykorzystują najczęściej logikę matematyczną, teorię relacji lub dziedziny pokrewne. Umożliwiają badanie struktury i semantyki procesów, przez co wydają się być wygodne do optymalizacji. Są na ogół niezależne od systemu cyfrowego i stosowalne z dowolną szczegółowością.

Programy w **językach programowania** umożliwiają badanie struktury i semantyki oraz określanie niezależnych fragmentów procesu. Są w ograniczonym stopniu niezależne od systemu cyfrowego.

Powyższa, z konieczności pobieżna, charakterystyka wskazuje, że żadna z omówionych metod nie posiada wszystkich postulowanych właściwości, a zatem nie nadaje się bezpośrednio do celów identyfikacji informacyjnej. Wydaje się jednak, że klasa metod algebraicznych jest najbliższa ideałowi i tam należy poszukiwać potrzebnego języka opisu.

* * *

Identyfikacja informacyjna nie eliminuje konieczności badania sprzętu liczącego i jego oprogramowania podstawowego, podobnie jak nie eliminuje konieczności poszukiwania związków zachodzących pomiędzy procesami informacyjnymi a systemami cyfrowymi. Stanowić ma ona pierwszy, niezbędny etap projektowania. Ostateczna synteza systemu informatycznego winna opierać się na:

- wynikach identyfikacji informacyjnej
- wynikach rozpoznania właściwości różnych systemów cyfrowych
- odpowiednich metodach doboru.

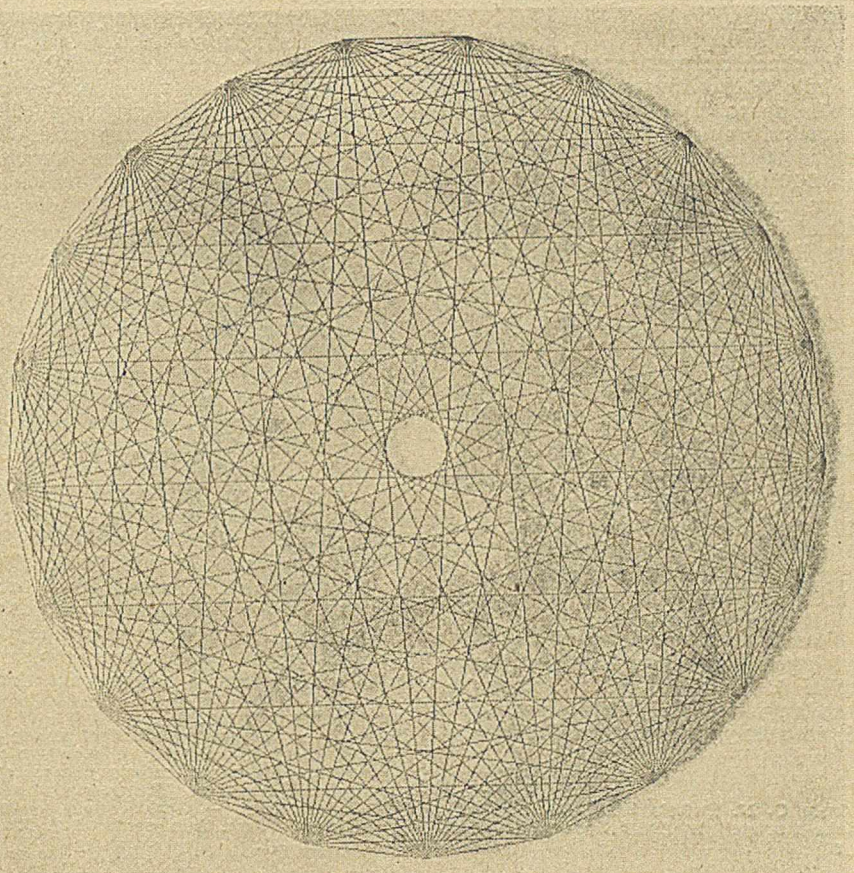
Wymienione wyżej elementy ściśle odpowiadają podziałowi ogólnego problemu syntezy systemów informatycznych.

LITERATURA

[1] Bromirski J. — Problemy projektowania systemów cyfrowych, INFORMATYKA nr 5/72

[2] Duch J. — Metody oceny systemów liczących, INFORMATYKA nr 1/74

[3] Forini di A. C. i in. — Informatyka. Problemy i perspektywy badań podstawowych w wielodyscyplinarnych dziedzinach, INFORNA Warszawa 1973



Optyczne pamięci cyfrowe*)

Ostatnie osiągnięcia w technice pamięci optycznych pozwalają sądzić, że ten rodzaj pamięci rokuje największe nadzieje w następnej generacji pamięci masowych.

Pamięci optyczne umożliwiają znaczną poprawę takich parametrów, jak pojemność i czas dostępu w stosunku do istniejących technik pamięciowych, jak dyski i taśmy magnetyczne. Opracowanie to stanowi przegląd rozwiązań pamięci tego typu.

Zainteresowanie pamięciami optycznymi ma swe źródło przede wszystkim w możliwości uzyskania tutaj znacznego zwiększenia gęstości zapisu, w stosunku do tradycyjnego zapisu magnetycznego. Zastosowany wymiar bitu może być tego rzędu, co długość fali świetlnej (0,5 μ). Głównym problemem pozostaje jednakże niezawodne adresowanie dużych obszarów ośrodka przechowującego informację, co jest niezbędne w celu osiągnięcia dużych pojemności wymaganych do pamięci masowej.

Istnieją dwa podstawowe systemy organizacji pamięci optycznej: tzw. system „bit po bicie" oraz pamięci obrazów holograficznych. Ten drugi system, prócz zwiększonej gęstości zapisu, pozwala na znaczne skrócenie czasu dostępu w stosunku do innych rodzajów pamięci masowych.

Oczekuje się, iż pamięci optyczne będą mogły skutecznie konkurować z innymi układami pamięciowymi o dużej pojemności w dziedzinie kosztów, pozwalając jednocześnie uzyskać korzystniejsze parametry pracy. Najdroższe w tych pamięciach są układy optyczne i dlatego koszt małych pamięci optycznych na bit przechowywanej informacji będzie bardzo wysoki, lecz wartość ta zmniejszy się znacznie przy systemach o dużej pojemności. Dlatego właśnie pamięci optyczne są przede wszystkim przeznaczone do systemów pamięci masowych.

Obecnie, układy na dyskach magnetycznych są w stanie przechowywać około $6 \cdot 10^9$ bitów w systemie składającym się z ośmiu zespołów dysków, z których każdy ma 20 powierzchni zapisu. Gęstość zapisu informacji jest tu rzędu 10^5 bit/cm² co daje skuteczną powierzchnię zapisu $6 \cdot 10^4$ cm². Aby powiększyć pojemność tych systemów do 10^{12} bitów należało by albo powiększyć sto kilkadziesiąt razy powierzchnię zapisu, albo taką samą ilość razy zwiększyć gęstość zapisu. Dotychczasowe doświadczenie wykazuje, że mechaniczne systemy adresowania powierzchni większych aniżeli $6 \cdot 10^4$ cm² są zawodne. Również podniesienie gęstości zapisu w żądanym stopniu jest przy obecnym stanie techniki zapisu magnetycznego nieosiągalne i dlatego trwają intensywne poszukiwania nowych technik, które pozwolą na realizację bardzo dużych pamięci.

Ograniczenie, jeśli chodzi o gęstość zapisu w pamięciach optycznych, wynika z dyfrakcji światła i w zakresie światła widzialnego wynosi w przybliżeniu 0,5 μ. Jeśli przyjąć, że odległość pomiędzy bitami jest porównywalna ze średnicą bitu, to uzyskana gęstość zapisu wynosi 10^8 bitów/cm².

W praktyce może okazać się niezbędnym mniej więcej dwukrotne powiększenie wymiaru bitu, lecz wciąż daje to gęstość zapisu $2 \cdot 10^7$ bitów/cm², co już zostało doświadczalnie zademonstrowane w laboratorium. Jeśli znów przyjąć, że mamy do dyspozycji powierzchnię rzędu $6 \cdot 10^4$ cm², to staje się oczywistym, że metody optyczne powinny być użyteczne przy realizacji pamięci o pojemności rzędu 10^{12} bitów. Możliwość osiągnięcia tak dużej pojemności została zademonstrowana przez firmę PRECISION INSTRUMENT Co. w laserowej pamięci archiwalnej UNICON 690 [1]), gdzie gęstość zapisu informacji jest rzędu $3 \cdot 10^6$ bitów/cm², a ośrodek pamiętający ma powierzchnię około $4 \cdot 10^5$ cm². Zapisu informacji dokonuje się za pomocą lasera, który wypala otwory w ośrodku pamiętającym, którym jest cienka warstwa metaliczna, natomiast odczyt oparty jest na wykrywaniu obecności tych otworów za pomocą wiązki laserowej o mniejszej mocy.

W konwencjonalnych pamięciach magnetycznych odległość pomiędzy głowicą zapisującą i ośrodkiem musi być porównywalna z najmniejszym wymiarem bitu. Na skutek tego wymagane są pływające głowice i pojawia się zagadnienie uszkodzeń głowic, szczególnie przy wyższych gęstościach zapisu. W pamięciach optycznych ta wysoka gęstość zapisu może być uzyskana bez umieszczania głowicy przetwornika lub końcowej soczewki obiektywu wyjątkowo blisko ośrodka zapisu. W pamięci systemu „bit po bicie" soczewki te mogą być odsunięte na odległość równą średnicy wiązki, to znaczy około 0,25 mm lub więcej od powierzchni ośrodka pamiętającego w celu wyeliminowania problemu uszkodzeń. Jednakże, głębia ostrości soczewki obiektywu jest bardzo ograniczona tak, że odległość od soczewki do powierzchni ośrodka musi być dokładnie regulowana, aby uniknąć kłopotów z rozdzielczością.

Dalszą cechą zapisu i odczytu optycznego jest proporcjonalność sygnału odczytu do natężenia laserowej wiązki odczytu, a ośrodek przechowujący potrzebny jest jedynie do modyfikacji natężenia, polaryzacji lub fazy wiązki w celu uzyskania sygnału użytecznego. Odróżnia to znacznie ten system od systemu magnetycznego, gdzie sygnał odczytu jest pochodną strumienia magnetycznego związanego z bitem i maleje ze zmniejszaniem się wymiarów bitu. Sygnał maksymalny przy zapisie optycznym jest, oczywiście, ograniczony przez maksymalną moc wiązki, która może padać na powierzchnię bitu bez spowodowania zniszczenia na skutek nadmiernego nagrzania ośrodka.

Holograficzne metody zapamiętywania optycznego mają te same zalety w stosunku do zapisu magnetycznego, a ponadto mają wiele własności dodatkowych. Po pierwsze, można tu zrealizować pamięci o względnie dużych pojemnościach (do $5 \cdot 10^9$ bitów) bez jakichkolwiek ruchów mechanicznych w systemie, o ile do odchylania wykorzystuje się bezinercyjny układ optyczny, a informacja ma postać stron, co pozwala na równoległe wprowadzanie i wyprowadzanie danych. Ta własność pozwala osiągać czasy dostępu poniżej 10 μs, w przybliżeniu o trzy rzędy wielkości mniej aniżeli graniczne wartości przy układach dyskowych. Wyeliminowanie urządzeń mechanicznych stwarza możliwość zwiększenia niezawodności. Ponadto, proces zapisu holograficznego dopuszcza pewien stopień tolerancji na kurz i wady ośrodka pamiętającego, ponieważ hologram nie jest obrazem informacji, lecz jej transformacją Fouriera. Ta własność jest bardzo ważna przy zapisie o dużej gęstości, ponieważ obszary pokryte cząstkami kurzu lub defekty

*) Na podstawie artykułu O. N. Tufte'a i D. Chena „Optical techniques for data storage" w IEEE Spectrum, no 2, vol. 10 z lutego 1973 r.

[1] Nowa wersja tej pamięci znana jest pod nazwą System 190. Jest to modułowa, usprawniona wersja UNICON 690 posiadająca niezmienione parametry, jeśli chodzi o gęstość zapisu i koszt za bit przechowywanej informacji.
J. H. Gilder — Optical data systems find a niche in the world of fast, fast computing. Electronic Design No 9/1974 str. 126—31.

w ośrodku pamiętającym mogą być znacznie większe aniżeli wymiar bitu.
Mimo tylu zalet pamięci optycznej w stosunku do obecnego stanu metod zapisu magnetycznego, pozostało tu wiele problemów, od których rozwiązania uzależniona jest możliwość realizacji pamięci o uprzednio wymienionych zaletach.

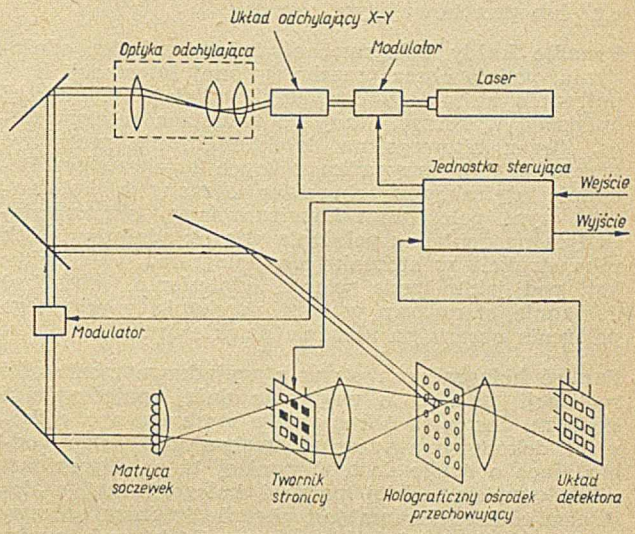
Realizacje pamięci optycznych

Jak już wspomniano, stosowane są obecnie dwa różne systemy pamięci optycznych. W metodzie „bit po bicie" wykorzystuje się energię wiązki laserowej zogniskowanej na ośrodku przechowującym w celu zapisania bitu informacji w postaci plamki. W danej chwili odbywa się zapis lub odczyt jednego bitu, zwykle stosuje się tu dostęp szeregowy, podobnie jak przy konwencjonalnym zapisie magnetycznym. Natomiast w technice wykorzystującej holograficzne metody przechowywania informacji mamy do czynienia z organizacją informacji w postaci stronic, zawierających od 10^3 do 10^5 bitów, które są zapisywane lub odczytywane jednocześnie. Możemy tu więc mówić o równoległym przechowywaniu lub odczycie danych zapamiętywanych na jednej stronicy. Systemy optyczne, oraz wymagania na ich elementy różnią się w przypadku obu tych organizacji. Poniżej przytoczony zostanie krótki opis obu tych systemów wraz ze streszczeniem wymagań stawianych podstawowym elementom tworzącym te systemy pamięciowe.

System „bit po bicie" składa się z laserowego źródła światła, modulatora optycznego włączającego i wyłączającego wiązkę oraz regulującego jej poziom natężenia, układu odchylającego, umieszczającego wiązkę w danym położeniu adresowym, ośrodka pamiętającego i optycznego układu detekcyjnego do odczytu z pamięci. Ponadto potrzebne są elementy optyczne do sterowania wymiarami wiązki, jak również do umieszczania jej i ogniskowania na ośrodku pamiętającym.
Podstawowymi zespołami pamięci optycznych są: system odchylający i ośrodek przechowujący, w którym można zapisać, odczytywać i wymazywać informację za pomocą tej samej lub podobnej wiązki światła. W przypadku pamięci typu archiwalnego informacja zapisywana jest tylko jeden raz i odczytywana wielokrotnie m. in. w celu zmniejszenia wymagań odnośnie do ośrodka pamiętającego.

Jako przykład procesu zapisu, odczytu i wymazywania w pamięci typu „bit po bicie" rozpatrzony zostanie przypadek mangetooptycznego ośrodka przechowującego w postaci warstwy MnBi. Warstwy te charakteryzują się jednoosiową anizotropią magnetyczną i są wstępnie magnesowe prostopadle do płaszczyzny warstwy. W celu zapisu bitu informacji, impuls o dużym natężeniu światła z lasera i modulatora odchylany jest do żądanego położenia i ogniskowany na małym obszarze (w przybliżeniu o średnicy jednego mikrona) ośrodka pamiętającego. Wielkie skoncentrowanie energii na tym obszarze powoduje nagrzanie jego powyżej punktu Curie. Gdy plamka ta oziębia się poniżej temperatury Curie, przykładane jest zewnętrzne pole magnetyczne o kierunku przeciwnym do namagnesowania pierwotnego, powodując odwrócenie namagnesowania plamki. Odpowiada to zapisowi jedynki w danym bicie. Proces wymazywania jest taki sam jak zapis, z tym, że zewnętrzne pole magnetyczne przykładane w czasie oziębienia ma przeciwny kierunek i przełącza kierunek namagnesowania do stanu pierwotnego. Do odczytu przechowywanego bitu wykorzystywany jest efekt magnetooptyczny. Gdy spolaryzowana wiązka laserowa przechodzi przez tę warstwę lub jest od niej odbijana, polaryzacja ta ulega obrotowi, a kierunek obrotu określony jest przez kierunek namagnesowania warstwy. Jeśli przed detektorem umieścimy polaryzator, zmiany kierunku polaryzacji zostaną przekształcone na zmiany natężenia, które mogą być odczytywane za pomocą fotodetektora.

W obecnie realizowanych pamięciach typu „bit po bicie" wymagane są jeszcze ruchome elementy mechaniczne, ponieważ aktualny poziom techniki odchylenia nie pozwala na objęcie dostatecznej liczby elementów.



Na rys. 1 przedstawione są schematycznie podstawowe podzespoły pamięci holograficznej. Proces pamiętania wymaga, aby bity informacji były najpierw wprowadzone do twornika stronicy, który jest binarnym modulatorem przestrzennym i składa się z umieszczonych w płaszczyźnie x-y zaworów świetlnych lub migawek sterowanych elektrycznie lub optycznie, zgodnie z wymaganymi położeniami i stanami bitowymi tak, jak to jest określone przez jednostkę sterującą. Transformuje to bity informacji na dwuwymiarowy cyfrowy obraz lub stronicę zawierającą jasne i ciemne punkty.

Twornik stronicy znajduje się w pobliżu soczewki i gdy oba te elementy oświetlone zostaną wiązką laserową, w płaszczyźnie holograficznego ośrodka pamiętającego utworzony zostanie obraz, który jest trasformacją Fouriera obrazu bitowego, wytworzonego przez twornik stronicy. Wiązka odniesienia, uzyskana przez rozdzielanie oryginalnej wiązki laserowej, nakłada się na ten obraz transformacji Fouriera. Wynikły stąd obraz interferencyjny zostaje zapisany w ośrodku pamiętającym w postaci hologramu. Hologramy przechowywane są również w innych położeniach na ośrodku pamięciowym poprzez przesuwanie padającej wiązki za pomocą układu odchylającego. Po układzie odchylającym występuje układ optyczny, który może skierować wiązkę do dowolnego elementu płaskiej matrycy soczewek. Przesunięcie wiązki z jednej soczewki do drugiej zmienia położenie zapisanego hologramu w płaszczyźnie x-y. Wiązka odniesienia musi oczywiście podążać za wiązką sygnałową tak, by obie padały na ten sam obszar ośrodka przechowującego informację.

Odczyt odbywa się przez podanie na hologram tylko wiązki odniesienia, lub innej wiązki padającej pod tym samym kątem co wiązka odniesienia. Wiązka ta ulega dyfrakcji na hologramie i obraz informacji z twornika stronicy podawany jest poprzez układ optyczny na układ fotodetektorów mający te same względne wymiary, co twornik stronicy. Każdy bit poprzednio przechowywany w tworniku stronicy pada na fotodetektor w tym układzie i informacja z powrotem jest przekształcana na sygnał elektryczny. Każdy fotodetektor połączony jest z przerzutnikiem tranzystorowym, który jest ustawiany przez sygnał z fotodetektora i odczytywany przez układ sterujący. Przechowywanie informacji w postaci hologramu zamiast rzeczywistego obrazu, jaki znajduje się w tworniku stronicy, ma istotne zalety. Po pierwsze, informacja odnośnie do każdego bitu w tworniku stroni-

cy rozłożona jest w całym hologramie. W wyniku tego hologram jest względnie nieczuły na zanieczyszczenia lub defekty ośrodka. Powodują one tylko niewielkie pogorszenie rozdzielczości i natężenia odczytu wszystkich bitów. Nie przesłaniają jednak poszczególnych bitów, jak to się dzieje w systemie „bit po bicie" lub przechowującym obrazy.

Ponadto każdy hologram, oświetlony wiązką odniesienia, podaje obraz przechowywanej informacji bezpośrednio na pojedyńczy układ detekcyjny. Układ optyczny, wykorzystywany do tworzenia rzeczywistego obrazu bitowego na detektorach, nie musi być przesuwany w przypadku różnych hologramów. Podobnie, dyfrakcyjny charakter obrazu na hologramie czyni położenie rekonstruowanego obrazu nieczułym na dokładne położenia wiązek odniesienia lub rekonstrukcji w stosunku do hologramu. Jednakże kąt, pod jakim pada wiązka odniesienia, musi być dokładnie sterowany w celu uzyskania poprawnego położenia obrazu na układzie detektorów.

Pamięć holograficzna stwarza wyjątkowe możliwości realizacji pamięci o względnie dużej pojemności i szybkim dostępie. Adresowanie może tu być całkowicie dokonywane przez niemechaniczne układy odchylające o czasie dostępu poniżej 10 μs. Ograniczeniem wymiaru takiej pamięci holograficznej z dostępem elektronicznym jest obecnie średnica soczewki końcowej, ponieważ we wszystkich tych systemach przynajmniej jedna soczewka musi być tak duża jak ośrodek zapisujący. Osiąga się tu gęstość zapisu do $6 \cdot 10^6$ bit/cm². Tak więc, przykładowo, maksymalnej średnicy soczewki 30 cm odpowiada pojemność $4{,}5 \cdot 10^9$ bitów. Pamięć o tej pojemności i czasie dostępu 10 μs stanowi obecnie dużą atrakcję.

Pamięci holograficzne stawiają specjalne wymagania odnośnie do podzespołów w nich stosowanych. Proponowana pojemność stronicy wynosi aktualnie 10^4 bitów. Przy zapewnieniu wystarczającej rozdzielczości średnica hologramu powinna wynosić ok. 1 mm. Ważnym parametrem jest tu moc lasera wymagana do zapisania hologramu o tych wymiarach. Ośrodek pamiętający o dużej energii zapisu będzie wymagał laserów o bardzo dużych energiach w impulsie do realizacji szybkiego zapisu. W pamięciach typu „bit po bicie" problem ten nie jest tak istotny, gdyż dokonuje się szeregowo i w danej chwili zapisuje się tylko jeden bit informacji na obszarze o średnicy około 1 μ.

Bardzo ważna jest również wydajność dyfrakcyjna ośrodka pamiętającego, określająca, jaka część promieniowania wiązki odczytowej zostaje ugięta, by wytworzyć zrekonstruowany obraz. Podczas odczytu energia wiązki dzieli się na wszystkie fotodetektory, a więc w przypadku stronicy składającej się z 10^4 bitów — na taką liczbę fotodetektorów. Jeśli wydajność dyfrakcyjna wynosi 0,01, a straty w układzie optycznym, modulatora i układzie dzielenia wiązki dodatkowo 10-krotnie zmniejszają jej natężenie, to tylko 10^{-7} natężenia początkowego wiązki będzie podać na każdy detektor. Ta znaczna redukcja musi być skompensowana przez powiększoną moc lasera w czasie odczytu oraz przez użycie bardzo czułych detektorów. Wydajność dyfrakcyjna różnych materiałów waha się od 10^{-4} do jedności, np. dla filmów fotograficznych sięga 6%. Istnieją też hologramy fazowe, gdzie obraz interferencyjny określony jest zmianą współczynnika załamania, a nie przepuszczalnością światła. Te hologramy mają wyższą wydajność dyfrakcyjną, sięgającą teoretycznie do 100% [2]).

[2]) W pierwszej połowie 1973 r. opracowano dwie doświadczalne pamięci holograficzne. Pierwsza, przez firmę RCA, o pojemności 10^6 bitów (1000 stronic po 10^3 bitów) i druga przez HARRIS INTERTYPE Corp. o pojemności $4 \cdot 10^5$ bitów (25 stronic po $16 \cdot 10^3$ bitów). Obie pamięci wykorzystują ośrodek termoplastyczny do przechowywania informacji, natomiast różnią się twornikami stronicy. W pierwszej pamięci jest on wykonany z ciekłego kryształu, a w drugiej — z ferroelektrycznego materiału ceramicznego. Podobny do tego ostatniego jest również twornik stronicy (z ceramiku PLZT) opracowany przez PLESSEY w Wielkiej Brytanii.

Materiały stosowane w pamięciach optycznych

Z braku miejsca wyliczone tu jedynie będą rodzaje materiałów, które stosuje się w pamięciach optycznych. Muszą one mieć własności fizyczne umożliwiające zapis, wymazywanie i odczytywanie informacji i być stabilnymi, pozwalając przechowywać zapisaną informację przez długi okres czasu, a wielokrotnie przeprowadzane wspomniane operacje zapisu, wymazywania i odczytu nie powinny zmieniać ich charakterystyk. Powinny też mieć strukturę wewnętrzną umożliwiającą gęsty zapis i pozwalać na pracę w normalnych warunkach otoczenia, temperaturze pokojowej i w powietrzu. Nieco zmienione wymagania dotyczą materiałów do pamięci archiwalnych i z wielokrotnym odczytem.

- Ośrodki termomagnetyczne były pierwszymi, które tu wykorzystano. Mówiło się wówczas o pamięci magnetooptycznej.
Rozróżnia się tutaj metodę zapisu przy punkcie Curie i metodę kompensacji temperaturowej.

- Ośrodki ferroelektryczne, charakteryzujące się histerezą polaryzacji elektrycznej i wykazujące efekt elektrooptyczny otwierają interesujące możliwości do zastosowań pamięciowych. Można tu wyróżnić układy typu ferroelektryk — fotoprzewodnik, materiały, w których powstają lokalne zmiany współczynnika załamania pod wpływem naświetlenia oraz inne układy ferroelektryczne.

- Materiały termoplastyczne mogą być wykorzystane w układzie dwuwarstwowym z warstwą fotoprzewodzącą lub też materiał termoplastyczny może być również fotoprzewodzącym. Wymazywanie informacji odbywa się tu oczywiście przez podgrzanie ośrodka.

- Materiały fotochromowe zmieniają swą charakterystykę absorbcji pod wpływem napromieniowania światłem o danej długości fali.

- Półprzewodniki amorficzne są od niedawna wykorzystywane w badaniach nad pamięciami optycznymi. Mechanizm pamiętania oparty jest na przejściu ze stanu amorficznego w krystaliczny.

Ocena kosztów

Trudno jest uzyskać dokładne wartości kosztów rozwijającej się dopiero techniki, jaką są pamięci optyczne, lecz pewne dane i ekstrapolacje pozwalają na ocenę rzędu wielkości. Pamięć archiwalna UNICON sprzedawana jest za ok. 10^{-4} centa/bit. Ponieważ koszt ośrodka przechowującego informację nie decyduje tutaj o koszcie całego systemu, można oczekiwać iż zastąpienie ośrodka archiwalnego ośrodkiem, gdzie informacja będzie mogła być zmieniana, nie zmieni zasadniczo kosztu za bit. Duże pamięci dyskowe o pojemności około 10^{10} bitów są sprzedawane obecnie po około $5 \cdot 10^{-3}$ centa/bit. i oczekuje się, że postęp w ciągu najbliższych kilku lat obniży tę wartość więcej, niż 5-krotnie. Dalsza obniżka kosztów przy użyciu konwencjonalnych metod będzie trudna. Szeregowe pamięci optyczne powinny początkowo być konkurencyjne w stosunku do systemów dyskowych o dużych pojemnościach (10^{11} — 10^{12} bitów). Jednakże dalsze usprawnienia i zmniejszenie kosztów techniki optycznej może umożliwić również opłacalność stosowania mniejszych systemów optycznych.

Pamięć holograficzna wymaga jeszcze usprawnień, które będą miały wpływ na koszt systemu. Ze względu na wątpliwości co do ostatecznego rozwiązania, trudno jest o ocenę kosztu. Jednakże szybki czas dostępu i wysoka szybkość przesyłania informacji uczynią pamięci holograficzne znacznie wydajniejszymi, aniżeli układy dyskowe w zakresie pojemności 10^8 — 10^9 bitów. I dlatego cena nie jest tu tak istotna, a koszt za bit może być porównywalnym. Decydować jednak będą wówczas korzystniejsze parametry funkcjonalne.

Tłum. i oprac. **Jan Ryżko**

Reguły prawidłowej eksploatacji i przechowywania pakietów dyskowych

Żadna ze współczesnych gałęzi przemysłu nie dorównuje przemysłowi komputerowemu w dynamice rozwoju, szybkości postępu technicznego i modernizacji wyrobów. Fakty te narzucają producentom elementów podstawowych, systemów komputerowych oraz nośników informacji szczególnie wysokie wymagania w zakresie jakości dostarczanych na rynek nowych wyrobów. Często podkreśla się jednak, że postęp ten nakłada również na kierownictwo i personel wykonawczy ośrodków obliczeniowych większe obowiązki w zakresie zapewnienia warunków prawidłowej eksploatacji sprzętu, gwarantujących uzyskanie właściwych efektów jego zastosowania.

Jest bezspornym faktem, że sprawność operacji przetwarzania danych zależy bezpośrednio od staranności operowania magnetycznymi nośnikami informacji oraz ich przechowywania. Pakiet dyskowy jest precyzyjnym instrumentem, w którym tolerancje dokładności wykonania są rzędu setnych części mikrona. Jest oczywistym, że w tej sytuacji operowanie dyskami wymaga szczególnej pieczołowitości. Codzienna praktyka, potwierdzona licznymi obserwacjami personelu serwisu technicznego producentów sprzętu komputerowego i magnetycznych nośników informacji wskazuje jednak, że zasada ta nie zawsze jest przestrzegana. Personel eksploatacyjny ośrodków komputerowych nie zawsze uświadamia sobie bowiem rzeczywiste koszty uzyskania zarejestrowanych danych oraz to, że środki zapobiegawcze podjęte w zakresie prawidłowego operowania i przechowywania nośników magnetycznych mogą dać oszczędności rzędu dziesiątków tysięcy dolarów, głównie w wyniku eliminacji powtórnych przebiegów przetwarzania na komputerze.

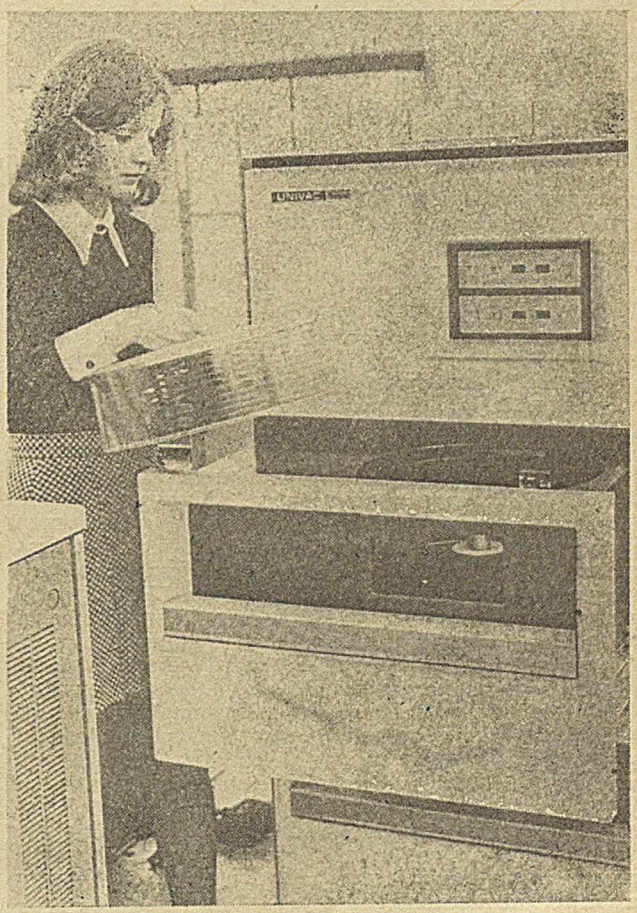
Poniższe wskazówki zostały oparte o materiały seminariów, jakie w wielu krajach firma 3M, producent pakietów dyskowych marki „SCOTCH" a zarazem pionier w dziedzinie magnetycznych nośników informacji, zorganizowała dla operatorów systemów komputerowych.

Operowanie

Opieka użytkownika nad pakietem dyskowym rozpoczyna się z chwilą otrzymania go od producenta. Pierwszą czynnością powinno być sprawdzenie opakowania przesyłkowego, czy nie wykazuje ono uszkodzeń, zwłaszcza zgniecenia narożników lub wgnieceń bocznych, świadczących o działaniu znacznej siły na zewnętrzne ścianki opakowania. Opakowanie to odporne jest na normalne warunki transportu, zabezpieczając pakiet przed kurzem, znacznymi zmianami warunków klimatycznych oraz fizycznymi uszkodzeniami wywołanymi np. wstrząsami. Gąbczaste wkładki wewnątrz opakowania spełniają rolę amortyzatorów przeciwwstrząsowych, zmniejszając wprawdzie siłę uderzenia o 2/3 lecz nie będąc w stanie całkowicie wyeliminować skutków ciężkich uszkodzeń. Jeżeli podejrzewa się uszkodzenie pakietu, należy go wyjąć z opakowania zewnętrznego i sprawdzić zasobnik plastikowy, czy nie ma on pęknięć lub wyszczerbień. Należy również zdjąć spodnią część zasobnika i sprawdzić, czy nie został uszkodzony dysk sektorowy *(Sector disc)*.

W niektórych jednostkach pamięci dyskowej istnieje możliwość spojrzenia na pakiet dyskowy od przodu. W takich przypadkach należy pakiet umieścić w jednostce i powoli puścić go w ruch obrotowy obserwując odstęp pomiędzy dyskiem wierzchnim i jego górną powierzchnią a dyskiem sektorowym i spodnią powierzchnią rejestrującą. Odstęp ten może być sprawdzianem wypaczenia spowodowanego uderzeniem pakietu od wierzchu lub od spodu. Po wprowadzeniu w ruch obrotowy pakiet należy szybko zatrzymać zanim nastąpi włączenie działania głowic. Obserwację powierzchni dysków należy przeprowadzać pod kątem występowania znaczniejszych odchyleń, porównując wzrokiem poziom położenia głowicy z poziomem płaszczyzny obracającego się dysku. Płaszczyzny wszystkich dysków powinny być równoległe w granicach tolerancji odchyleń wynoszącej ok. 0,3 mm w stosunku do płaszczyzny odniesienia. Jeżeli odchylenie płaszczyzny powierzchni dysku przekracza powyższą granicę tolerancji, to istnieje duże prawdopodobieństwo powstania kolizji głowicy z dyskiem, stwarzającej możliwość poważnego zatarcia powierzchni magnetycznej przez głowicę *(head crash)*.

Jeżeli istnieje przypuszczenie, że pakiet został uszkodzony w transporcie wskutek wypaczenia powierzchni dysków albo w wyniku drgań mechanizmu napędowego jednostki pamięci, operator komputera pakietu tego w żadnym przypadku nie powinien używać do chwili jego sprawdzenia przez specjalistę z serwisu producenta. Tę samą zasadę stosuje się do przypadków uszkodzenia pakietów wskutek nieuważnego operowania podczas eksploatacji. Operatorzy powinni zawsze przyznawać się do spowodowanych przez nich uszkodzeń pakietów i obowiązkowo przekazywać je do sprawdzenia, nie ryzykując powstania znacznych strat finansowych wskutek zniszczenia danych oraz konieczności zużycia czasu komputera na ich odtworzenie. Nawet tak odporna powłoka magnetyczna, jaką charakteryzuje się pakiet typu „*Crashguard*" („odporny na zatarcie") produkcji firmy 3M, nie uchroni



całkowicie danych w przypadku poważnego uszkodzenia dysku.

Inną potencjalną przyczyną kolizji głowicy z powierzchnią magnetyczną dysku (hdi = *head do disc interference*) są zanieczyszczenia pochodzące z powietrza. Jeżeli pakiet dyskowy jest w zasobniku albo gdy znajduje się on w jednostce pamięci dyskowej, to istnieje minimalna możliwość jego zanieczyszczenia. Większość obecnie produkowanych pakietów ma filtry wbudowane w podstawę ich konstrukcji. Przed umieszczeniem pakietu w jednostce pamięci zaleca się przetarcie filtru celem upewnienia się, czy nie jest on nadmiernie zanieczyszczony lub pęknięty. Jeżeli bowiem filtr jest zanieczyszczony, to zmniejsza on obieg powietrza w pakiecie, redukując tym samym wysokość poduszki powietrznej pod głowicami. Powoduje to niszczenie powłoki magnetycznej dysków, której cząsteczki powodują szybkie zatykanie kanalików pochłaniacza oraz zanieczyszczenie całej powierzchni filtru.

Wszystkie te czynniki znacznie zwiększają prawdopodobieństwo kolizji głowicy z powierzchnią dysku a w konsekwencji zarysowania warstwy magnetycznej oraz zbierania się resztek tlenku na głowicy. Proces ten ma charakter kumulacyjny: im większe jest nawarstwianie się resztek tlenku na głowicy, tym szybciej prowadzi to do zatarcia przez nią powierzchni dysku. Jeżeli nie zostanie to odpowiednio wcześnie wykryte, zanieczyszczona głowica będzie powodowała niszczenie powierzchni wszystkich pakietów kolejno wprowadzanych do tej jednostki pamięci, a z kolei pakiety te będą zanieczyszczały głowice innych jednostek dyskowych, do których zostaną wprowadzone. Dlatego podstawowym warunkiem prawidłowej eksploatacji pakietów dyskowych jest stałe czuwanie nad czystością filtrów.

Wymiana zanieczyszczonego filtru jest czynnością bardzo prostą: pierścień „0" należy ścisnąć pomiędzy kciukiem a palcem wskazującym aż wypadnie on z rowka zaciskowego. Należy wówczas podłożyć mały śrubokręt lub podobne narzędzie pod pierścień, który należy oddzielić od filtra wypychając go ku górze. Stary filtr należy wyjąć i zastąpić nowym, a następnie go zamocować wprowadzając do rowka zaciskowego pierścień „0". Zarówno nowy filtr, jak i pierścień „0" przed zamocowaniem powinny być sprawdzone, czy przypadkiem nie są uszkodzone. Odnosi się to zwłaszcza do pierścienia ze względu na możliwość jego uszkodzenia podczas operacji wyjmowania z pakietu. Wymiana filtru nie powoduje żadnej przerwy w eksploatacji pakietu, ponieważ decydującym czynnikiem są w tym zakresie warunki klimatyczne pomieszczeń, z których jednostka pamięci dyskowej pobiera powietrze.

W normalnych warunkach eksploatacji nie ma potrzeby regularnego czyszczenia powierzchni poszczególnych dysków, ponieważ nie powinny one podlegać jakimkolwiek zanieczyszczeniom. Natomiast wystawienie odsłoniętego pakietu na warunki sali komputera spowoduje więcej zanieczyszczeń wskutek istniejącego tam kurzu niż można by usunąć w wyniku najbardziej starannych zabiegów oczyszczających. Jeżeli jednak czyszczenie powierzchni dysków okaże się konieczne, to powinno być ono wykonane przez osobę o odpowiednich kwalifikacjach. Natomiast często może być potrzebne czyszczenie zewnętrznej powierzchni ochronnych zasobników typu LEXAN, używanych do osłony pakietów dyskowych marki SCOTCH, ponieważ zasobniki te jako wyroby z tworzywa sztucznego przyciągają kurz wskutek działania sił elektrostatycznych. Do tego celu wystarcza użycie wilgotnej ściereczki. Okresowo należy również sprawdzać piastę pakietu pod kątem prawidłowości przenoszenia na nią smaru z wrzeciona mechanizmu napędowego. Nadmiar smaru może bowiem spowodować w pakiecie wprowadzonym do jednostki pamięci dyskowej albo jego lekkie nachylenie albo wytworzenie się rodzaju hermetycznej uszczelki wokół piasty, utrudniającej znacznie zdejmowanie pakietu z wrzeciona mechanizmu napędowego. Smar ten można łatwo usunąć z piasty przecierając ją ściereczką nawilżoną alkoholem izopropylowym.

Jedną z istotnych własności pakietów dyskowych marki SCOTCH jest tzw. wyważenie dwupłaszczyznowe *(dual-plane balancing)*, które eliminuje drgania pionowe oraz obrotowe. Wyważenie to zapewnia odpowiednie ustawianie ciężarków wyważających, znajdujących się na wierzchu i spodzie pakietu. Ciężarek spodni jest łatwiej dostępny ale nie powinien być ruszany. Usunięcie tego ciężarka powoduje naruszenie wyważenia, a w konsekwencji możliwość powstania drgań oraz zniszczenia zarejestrowanych na dysku danych. Podczas wkładania pakietu do zespołu napędowego należy upewnić się czy pokrywa pakietu obraca się tak, że można ją łatwo przekręcić aż do oporu. Brak zamocowania pakietu może spowodować w czasie ruchu powstawanie wibracji, natomiast nadmierne zaciśnięcie uszkadza zespoły wrzeciona, zarówno w mechaniźmie napędowym, jak i w pakiecie. Pokrywę zasobnika należy zdejmować z pakietu zawsze pionowo, podobnie zresztą jak przy operacji nakładania. Zdejmowanie i nakładanie pokrywy zasobnika pakietu pod kątem jest główną przyczyną naginania powierzchni dysków nr 1, 3 i 5, co niestety zdarza się w praktyce eksploatacyjnej jeszcze bardzo często.

Podczas operacji wymiany pakietu w jednostce pamięci nie należy nigdy próbować zatrzymywać obracającego się pakietu ręką. Czas zatrzymania ruchu obrotowego pakietu jest sterowany przez mechanizm napędowy, co warunkuje minimalne zużycie łożysk wrzeciona oraz znacznie redukuje momenty obrotowe przenoszone stąd do poszczególnych części pakietu. Ręczne zatrzymywanie pakietu jest niebezpieczne również dla operatora, któremu obracający się pakiet może pokaleczyć ręce. Wymiana spodniej części zasobnika powinna zawsze odbywać się przez jej nałożenie na pakiet a nie odwrotnie, ponieważ stożek pakietu mógłby spowodować przebicie filtru.

Końcowym, wartym podkreślenia zaleceniem w zakresie prawidłowej eksploatacji pakietów dyskowych jest problem użycia etykiet. Samoprzylepne etykiety umieszczane we wgłębieniu tarczy zaciskowej zasobnika zalecane są tylko wtedy, gdy zastosowano w nich klej nieprzenoszący. Etykiety pozostawiające po ich zdarciu resztki kleju mogą być istotnym źródłem zanieczyszczeń. Etykiet nie należy również nigdy nalepiać jedną na drugiej, ponieważ prześwit pomiędzy tarczą zaciskową a pokrywą dysku jest bardzo mały a z upływem czasu ma tendencję do zdzierania się powodując powstawanie zanieczyszczeń. Zjawisko to występuje jeszcze ostrzej w przypadku wytłaczanych etykiet plastikowych, które są grubsze, zawijają się na krawędziach i są źródłem kurzu powstającego ze startego plastiku. Etykiety te mają również określony ciężar, który może wpłynąć na zachwianie istniejącego wyważenia pakietu, co nakazuje unikanie ich stosowania.

Pakiety przystosowane do stosowania w urządzeniach kompatybilnych z jednostką pamięci dyskowej IBM 3330 mają w górnej części pokrywy specjalne wgłębienie przeznaczone do umieszczania etykiet plastikowych. W tym przypadku jedynym niezbędnym zaleceniem jest zapewnienie prawidłowego a zwłaszcza trwałego przytwierdzenia etykiety plastikowej we wspomnianym wgłębieniu pokrywy.

Przechowywanie

Pakiety dyskowe powinny być zawsze przechowywane w zasobnikach ustawionych na poziomej płaszczyźnie. Zasobnik powinien mieć zawsze swą dolną część, która zabezpiecza pakiet przed zanieczyszczeniem. Ze względu na niebezpieczeństwo uszkodzenia dysku sektorowego, pakietów nigdy nie należy układać jeden na drugim. Do przenoszenia pakietów z miejsca na miejsce należy używać wyłącznie uchwytów, które podtrzymują je w ich środkach ciężkości. W żadnym przypadku pakietu nie można trzymać za jego podstawę przy braku dolnej części zasobnika, ponieważ wywierany wtedy nacisk spowoduje wypa-

czenie płaszczyzny dysku sektorowego oraz uszkodzenie dolnej powierzchni rejestrującej.
Dyski przystosowane są do eksploatacji w temperaturze od 15,5 do 32,2°C oraz zakresu 8—80% wilgotności względnej, z wyłączeniem warunków mogących spowodować skroplenie pary wodnej. Przechowywanie dysków może odbywać się w temperaturze od —40 do +65,5°C przy wilgotności względnej 8—80%. W przypadkach, gdy przechowywanie odbywało się w warunkach wykraczających poza wyżej podane graniczne wartości warunków eksploatacji, pakiety należy przez okres 24 godzin przed terminem ich użycia poddać sezonowaniu. W przypadkach przechowywania przez dłuższy okres czasu, pakiety dyskowe najlepiej jest składować w ich oryginalnych opakowaniach transportowych, co w pełni zabezpiecza je przed kurzem, dużymi zmianami warunków atmosferycznych oraz fizycznymi uszkodzeniami.

Dyski kasetowe i elastyczne

Pojedyncze dyski kasetowe mają bardzo odporną konstrukcję i tym samym są niewrażliwe na rodzaje uszkodzeń wymienionych przy omawianiu pakietów dyskowych. Jedynymi zaleceniami jest unikanie w czasie użytkowania stłuczenia części z tworzyw sztucznych, takich jak uchwyt lub obudowa otworu wejściowego, potrząsanie dyskiem oraz stłuczenie wskutek upuszczenia go na podłogę.

Ostatnia nowość w grupie dysków, a mianowicie dysk elastyczny *(floppy disc)* zwany również dyskettą *(diskette)*, wymaga bardziej strannego obchodzenia się, zwłaszcza przy wyjmowaniu go z zespołu napędowego. W czasie operowania należy zważać, aby za wszelką cenę uniknąć wygięcia lub wypaczenia. Dyskietta w czasie gdy nie jest użytkowana powinna być przechowywana w kopercie oraz w zasobniku zabezpieczającym ją przed zatarciem lub innym uszkodzeniem.

Na podstawie materiałów informacyjnych firmy 3M opracował **W.K.**

PRZEGLĄD WYDAWNICTW

Bibliografia wydawnictw polskich z dziedziny informatyki

● **Analiza obwodów prądu** zmiennego (program ALIZ i przykłady) — ROMANOWSKI H. Wyd. Politechniki Śląskiej im. W. Pstrowskiego, Gliwice 1973 r., s. 70, cena zł 5,—. Seminarium obwodów elektrycznych. Z. 2.

Sposób posługiwania się programem ALIZ: zastosowanie, wprowadzanie danych, rodzaje obliczeń. Opis programu ALIZ: opis procedur użytych w programie ALIZ, program ALIZ i poprawki „charakterystyka", przykłady.
Materiały przeznaczone są dla studentów Wydziału Elektrycznego Politechniki Śląskiej, innych wyższych szkół technicznych oraz inżynierów projektantów.

● **Informator o systemach powielarnych.** Cz. 1. Wyd. Towarzystwa Naukowego Organizacji i Kierownictwa, Szczecin 1974, s. 235, INFOGRYF 74. Giełda systemów.
Systemy technicznego przygotowania i planowania produkcji: system projektowania procesów przetwarzania BITYP, decyzyjny system dynamicznego planowania produkcji podstowowej, pomocniczej i środków towarzyszących ESPER, system planowania i przygotowania produkcji budowlano-montażowej dla Kombinatu Budowy Domów, program obliczania składników planu operatywnego w przedsiębiorstwie budowlano-montażowym itd. Systemy gospodarki materiałowej: podsystem EPD „Gospodarka materiałowa", system limitowania i rozliczania zużycia materiałów, system kontroli realizacji produkcji i zużycia materiałów w fabryce domów itd. Materiały przeznaczone są dla projektantów odcinkowych systemów informatycznych.

● **Informtor o systemach powielarnych.** Cz. 2. Wyd. Towarzystwa Naukowego Organizacji i Kierownictwa, Szczecin 1974, s. 221, INFOGRYF 74. Giełda systemów.
Systemy zatrudnienia i płac: system ewidencji osobowej, system rozliczania pracochłonności i funduszu płac, system ewidencji i analizy kadr, system rozliczeń płacowych pracowników dniówkowych i umysłowych. System gospodarki wyrobami gotowymi: system ewidencji i rozliczeń wyrobów gotowych, system planowania i rozliczania umów, produkcji i zbytu wyrobów gotowych DANA 1. System gospodarki środkami trwałymi: system gospodarki środkami trwałymi OBIEKT, system ewidencji środków trwałych i rozliczenie amortyzacji SARST. Systemy normatywnego rachunku kosztów: rachunek kosztów normatywnych w przedsiębiorstwie transportowym budownictwa, rachunek kosztów normatywnych przedsiębiorstwa budowlano-montażowego.

Materiały przeznaczone są dla projektantów odcinkowych systemów informatycznych.

● **Informator o systemach powielarnych.** Cz. 3. Wyd. Towarzystwa Naukowego Organizacji i Kierownictwa, Szczecin 1974, s. 303, INFOGRYF 74. Giełda systemów.

SKOB — Skomputeryzowana kartoteka obiektów, SAK 71. System kosztorysowania, SKORMEL — System kosztorysowania robót melioracyjnych, MERSPED — System przetwarzania informacji o środowisku glebowo-przyrodniczym dla potrzeb inwestycji wodno-melioracyjnych, SAROTL — System automatycznego rozliczania obrotu towarowego lekami, ESRODOM 1 — System rozliczania dostawców usług telekomunikacyjnych. SART — System automatycznego rozliczania usług telekomunikacyjnych, WEKTOR, GWIP — System gromadzenia i wyszukiwania informacji patentowych itd.

Materiały przeznaczone są dla projektantów informatycznych systemów odcinkowych.

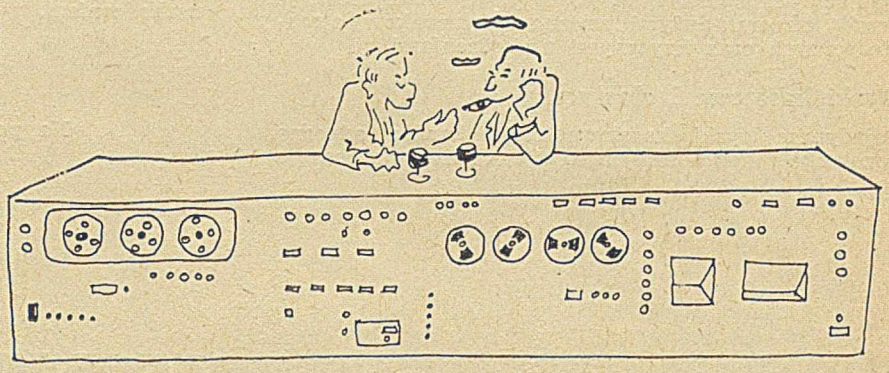
● **Obsługa operatorska elektronicznych maszyn cyfrowych ODRA 1300 i ICL 1900** — KLIMKIEWICZ S. Wyd. Głównego Urzędu Statystycznego, Warszawa 1974, s. 142.

Charakterystyka EMC. Obsługa operatorska systemów operacyjnych. Translatory języków programowania. Podstawowe programy standardowe. Załączniki.

Podręcznik przeznaczony jest dla operatorów obsługujących EMC, eksploatowane w Ośrodkach Elektronicznych Głównego Urzędu Statystycznego.

J. Klamborowski

dalszy ciąg bibliografii znajduje się na str. 52



Oznaczanie cyfry „0" i litery „O"

Elektroniczna maszyna cyfrowa z każdym rokiem coraz silniej wkracza do wszystkich dziedzin naszego życia. Jednocześnie w praktyce jej stosowania zaczynają występować różnego rodzaju problemy wynikające często z rozbieżności interpretowania lub wręcz niedoceniania obowiązujących ustaleń czy instrukcji.

W takich warunkach powstały u nas w praktyce trzy różne metody oznakowania i tym samym rozróżniania cyfry „zero" i litery „O":

Pierwsza
— cyfrę „zero" piszemy 0
— literę „O" piszemy Ø

Druga
— cyfrę „zero" piszemy Ø
— literę „O" piszemy ⊖

Trzecia
— cyfrę „zero" piszemy Ø
— literę „O" piszemy O

Oprócz tego spotkać można jeszcze inne wersje ukośnego lub poziomego przekreślania cyfry „zero" i litery „O". Nietrudno się domyśleć skutków tak różnorodnego oznakowania. Jeśli np. na taśmie dziurkowanej jest napisana nazwa programu # ZOKA, to nigdy nie wiadomo, czy oznacza to „Z — zero — KA", czy też „ZOKA". I podobnie, jeśli jest napisane: # ZØKA, ta może to oznaczać zarówno „Z — — zero — KA", jak i „ZOKA".

Rozbieżność interpretacji symboli powoduje błędną perforację kart lub taśmy papierowej i szczególnie dużo kłopotu sprawia przy uruchamianiu programów, gdyż wówczas zachodzi sumowanie błędów perforacji z błędami programowania. Wyszukiwanie błędów i ich korekta powoduje wydłużanie się czasu uruchamiania programów oraz znaczny wzrost kosztów, głównie w wyniku dodatkowego zużycia czasu komputera.

Co na ten temat pisze obowiązująca literatura?

Podręcznik języka PLAN dla komputera ODRA 1304, wydany przez Zakłady ELWRO w rozdziale 2 na stronie 4 podaje, że należy stosować następujące zasady:

— cyfrę „zero" piszemy jako 0
— literę „O" piszemy jako Ø

Odpowiadające mu wydawnictwo zagraniczne, zatytułowane PLAN CODING i opublikowane przez firmę ICL w tłumaczeniu polskim zawiera na ten temat następującą informację:

„...na arkuszu programowym „O" w słowie „COKE" jest przekreślone przez środek kreską poziomą. Należy to robić zawsze, aby uniknąć pomylenia litery „O" z cyfrą „0", które mają w maszynie inną postać...".

Z powyższego można wysnuć jednoznaczny wniosek, że zarówno literatura krajowa, jak i zagraniczna jest zgodna co do tego, że cyfrę „zero" piszemy zawsze jako 0, natomiast literę „O" piszemy jako Ø. Pisząc więc programy komputerowe oraz przygotowując dane, powinniśmy stale pamiętać o używaniu znaku 0 dla cyfry „zero" bez przekreślania, natomiast dla litery „O" — z przekreśleniem.

Szersze spopularyzowanie, a co ważniejsze — konsekwentne przestrzeganie jednolitych oznaczeń w tym zakresie byłoby bardzo istotnym elementem wzrostu efektywności zastosowań komputerów.

Zygmunt Topolewski
Wrocław

Romuald JAGIELSKI
Rafineria Nafty GDAŃSK

681.322.06:681.327.63

Operowanie danymi w DOS JS EMC

Przedstawiono sposoby sterowania operacjami wejścia/wyjścia w DOS Jednolitego Systemu EMC, metody organizacji zbiorów oraz możliwości stosowania języków wyższego rzędu PL/1, FORTRAN i RPG.

Dyskowy System Operacyjny Jednolitego Systemu (DOS JS) zawiera zbiór programów, przeznaczonych do operowania danymi, zwany Systemem Sterowania Wejściem/Wyjściem (SSWW). Na SSWW składają się programy dwóch typów:

- Fizyczny System Sterowania Wejściem/Wyjściem (Fizyczny SSWW)
- Logiczny System Sterowania Wejściem/Wyjściem (Logiczny SSWW).

Fizyczny SSWW stanowi część programu nadzorczego *(Superwisor)*. Zadaniem jego jest inicjowanie oraz kontrola operacji WE/WY, analiza błędów i wyjątkowych sytuacji w urządzeniach WE/WY, obsługa przerwań WE/WY.

Logiczny SSWW korzysta z programów Fizycznego SSWW, a ponadto zapewnia łączność z programami użytkownika, organizację i przetwarzanie zbiorów danych.

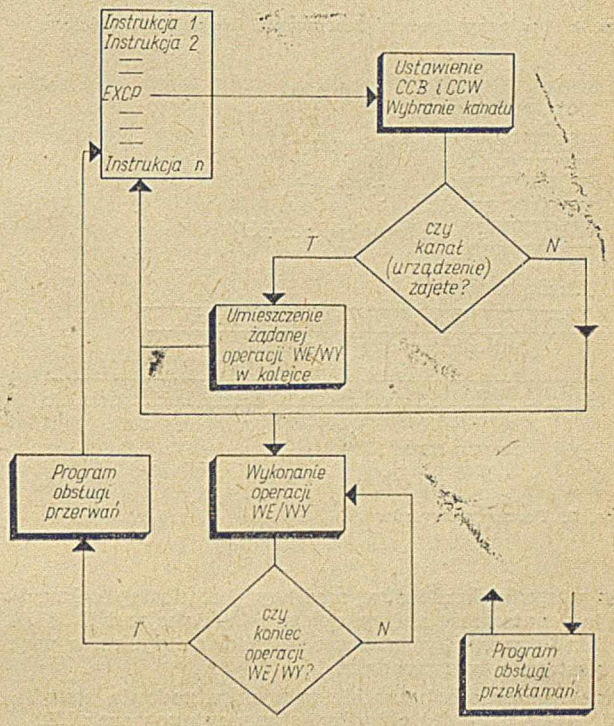
Sterowanie operacjami WE/WY

Fizyczny SSWW jest uruchamiany przez makroinstrukcję EXCP, która inicjuje wykonywanie programu kanałowego. Makroinstrukcja EXCP odsyła z kolei do makroinstrukcji CCB, tworzącej **blok sterowania rozkazami** (CCB) w kanale (rys. 1).

Fizyczny SSWW daje programiście duże możliwości pisania efektywnych programów, lecz wymaga dokładnej znajomości urządzeń i systemu operacyjnego oraz pisania własnych pracochłonnych programów kanałowych.

Logiczny **SSWW** składa się z modułów operujących zbiorami danych zdefiniowanych w programie użytkownika. Dane o zbiorach umieszcza programista w następujących makroinstrukcjach typu DTF *(Define The File)*:

DTFCN — zdefiniować zbiór dla monitora
DTFCD — zdefiniować zbiór na kartach
DTFPR — zdefiniować zbiór dla drukarki



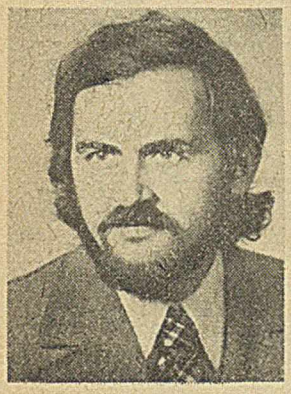
Rys. 1. Fizyczny System Sterowania Wejściem/Wyjściem

DTFMT — zdefiniować zbiór na taśmie magnetycznej
DTFSD — zdefiniować zbiór sekwencyjny na dyskach
DTFDA — zdefiniować zbiór losowy na dyskach
DTFIS — zdefiniować zbiór indeksowo sekwencyjny na dyskach

DTFDI — zdefiniować zbiór niezależny od urządzenia
DTFPT — zdefiniować zbiór na taśmie dziurkowanej.

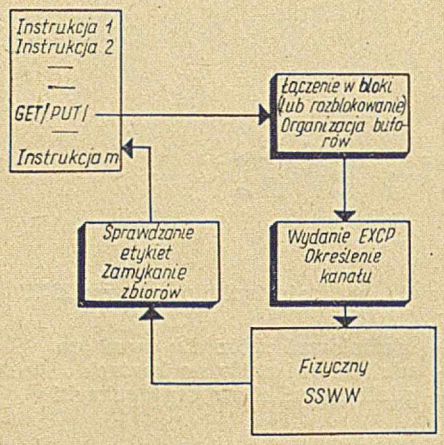
Logiczny SSWW używa programów Fizycznego SSWW do przesyłania danych (rys. 2), jednakże programista zwolniony jest z obowiązku pisania uciążliwych programów kanałowych.

Makroinstrukcje tworzą tak zwane tablice DTF, które zawierają charakterystyczne dane o określonym zbiorze. Każdej instrukcji DTF musi towarzyszyć instrukcja operatywna xxMOD (np. CDMOD — moduł zbioru na kartach), uruchamiająca logiczny moduł SSWW. Jest to podprogram systemowy, do którego należy: czytanie i zapisywanie danych, kontrola sytuacji wyjątkowej, łączenie zapisów (rekordów) w bloki, wydzielanie zapisu z bloku, umieszczanie zapisu w obszarze roboczym itd. Odpowiednie moduły są włączane

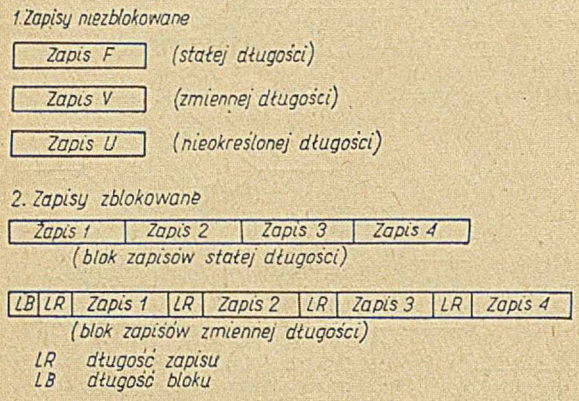


Mgr inż. ROMUALD JAGIELSKI ukończył studia o specjalności maszyny matematyczne na Politechnice Charkowskiej (1967 r.). Pracował jako konstruktor urządzeń peryferyjnych, a następnie jako projektant systemów EPD. Pełnił funkcję kierownika Zakładu Obliczeniowego Instytutu Maszyn Matematycznych Oddział w Toruniu (1972—1974) gdzie brał udział w pracach nad Jednolitym Systemem EMC. Obecnie jest kierownikiem Zakładowego Ośrodka Elektronicznej Techniki Obliczeniowej Rafinerii Nafty GDAŃSK (w budowie), oraz pracownikiem naukowo-badawczym Instytutu Matematyki Uniwersytetu Gdańskiego. Ma otwarty przewód doktorski w Katedrze Cybernetyki Politechniki Kijowskiej.

do programu użytkownika, przy czym programista może translować DTF i logiczne moduły razem ze swoim programem lub oddzielnie, przewidując ich późniejsze łączenie za pmocą programu REDAKTOR *(Linkage Editor)*.



Rys. 2. Logiczny System Sterowania Wejściem/Wyjściem



Rys. 3. Formaty zapisów

Do wszystkich zbiorów możliwy jest dostęp sekwencyjny, natomiast zbiory opisane instrukcjami DTFDA i DTFIS mogą być utworzone tylko na dysku, a więc z dostępem bezpośrednim.

Przetwarzanie odbywa się w wyniku podania instrukcji PUT i GET dla zbiorów sekwencyjnych oraz READ i WRITE dla zbiorów o dostępie bezpośrednim.

W DOS stosuje się zapisy (rys. 3) o następujących długościach: stałej (F — wszystkie zapisy mają jednakową długość), zmiennej (V — zapisy mają różne, ale zawsze możliwe do określenia długości), nieokreślonej (U — długość zapisu jest dowolna, nie można też jej dokładnie przewidzieć).

W celu zwiększenia szybkości przesyłania danych, zapisy można grupować w bloki (oprócz zapisów typu U).

Metody organizacji zbiorów

Zbiory zorganizowane sekwencyjnie zawierają zapisy uporządkowane w ściśle określony sposób (w takiej kolejności, w jakiej zostały zapisane). Zapisy — jak już wspomniano — mogą być stałej, zmiennej lub nieokreślonej długości, blokowane lub nieblokowane. Organizacja ta jest typową dla pamięci taśmowych, ale zbiory sekwencyjne mogą być tworzone również na dysku. Następujące instrukcje DTF definiują zbiory sekwencyjne: DTFCD, DTFTP, DTFPR, DTFMT, DTFSD, DTFDI. Dostęp do zapisów umożliwiają rozkazy GET i PUT, które pozwalają na jednoczesne przesyłanie danych i ich przetwarzanie.

Stopień nałożenia się tych operacji w czasie zależy od liczby wybranych obszarów wejścia/wyjścia, obecności bufora w urządzeniach WE/WY i obszaru roboczego (rys. 4).

Podczas operacji WE/WY, zapisy lub bloki zapisów wprowadzane są do obszarów pamięci operacyjnej. DOS dopuszcza użycie jednego lub dwóch obszarów WE/WY oraz jednego obszaru roboczego. Tabela I podaje różne warianty organizacji WE/WY i uzyskiwane przy tym efekty.

Organizacja sekwencyjna wskazana jest dla zbiorów dużych, gdy czas dostępu nie jest istotny, a częstotliwość zmian w zbiorze duża.

W zbiorach o **organizacji indeksowo-sekwencyjnej** zapisy uporządkowane są w zbiorze według wzrastającej wielkości klucza. Klucz jest polem 1—255 znakowym, poprzedzającym każdy zapis lub blok danych. Zbiór zapisany jest w trzech obszarach: obszar główny danych, obszar nadmiaru, indeksy.

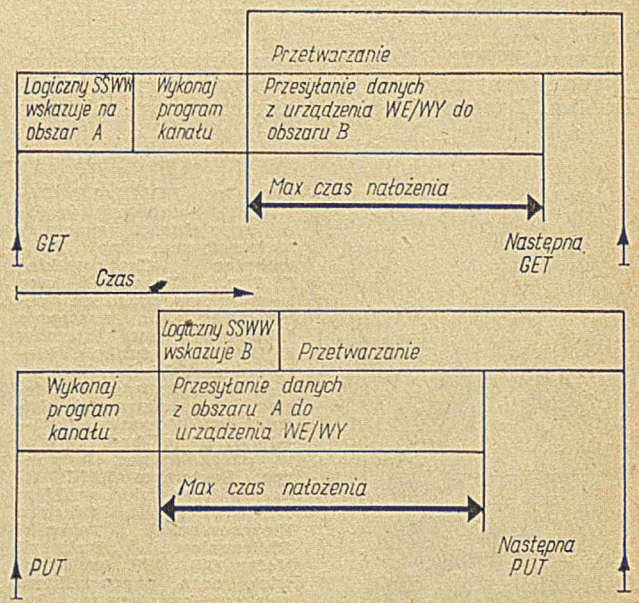
W obszarze głównym umieszcza się zapisy podczas zakładania zbioru. Gdy zachodzi potrzeba uzupełnienia zbioru, a nie ma już miejsca w obszarze głównym, nowe zapisy rejestruje się w obszarze nadmiaru, dzięki czemu nie zachodzi konieczność przepisywania całego zbioru.

Struktura zbiorów indeksowo-sekwencyjnych (rys. 5) opisana jest za pomocą dwóch lub trzech indeksów. Są to: indeks główny, indeks cylindrów i indeksy ścieżek. Indeks główny wykorzystuje się tylko przy bardzo dużych zbiorach, natomiast pozostałe indeksy występują zawsze. Każdemu zbiorowi odpowiada indeks cylindra, a każdemu cylindrowi — indeks ścieżki. Indeksy umożliwiają szybki dostęp do poszczególnych zapisów przy bezpośredniej metodzie dostępu lub przetwarzania zapisów w kolejności wzrastania klucza (sekwencyjnie).

Przy przetwarzaniu wyrywkowym programista podaje klucz, wskazując tym samym zapis, który ma być przetwarzany.

Przy przetwarzaniu sekwencyjnym użytkownik wskazuje zapis, od którego zaczyna się sekwencyjne przetwarzanie następnych zapisów.

W zbiorach indeksowo-sekwencyjnych stosuje się wyłącznie zapisy długości stałej. Każdy zapis musi mieć klucz. Używać można tylko standardowych etykiet zbiorów. Przetwarzanie rozpoczynają instrukcje PUT i GET (dostęp sekwencyjny) lub READ i WRITE (dostęp bezpośredni). Zbiór należy opisać instrukcją DTFIS.

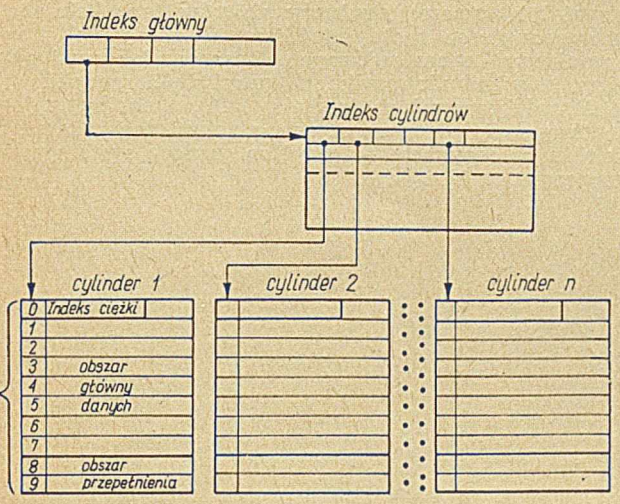


Rys. 4. Jednoczesność wykonywania operacji WE/WY i przetwarzania (nakładania się operacji) przy dwóch obszarach WE/WY (A i B) bez obszaru roboczego dla zapisów niezblokowanych

Tabela I

Efektywność jednoczesnego wykonywania operacji WE/WY i przetwarzania

Zapisy	Liczba obszarów WE/WY	Obszar roboczy	Efektywność nałożenia (overlap) operacji WE/WY i przetwarzania
niebloko-wane	1	nie ma	Nałożenie tylko dla buforowanych urządzeń WE/WY. Nie ma nałożenia dla pamięci taśmowych i dyskowych.
		jest	Nałożenie dla każdego zapisu
	2	nie ma	Nałożenie dla każdego zapisu
		jest	Nałożenie dla każdego zapisu. Obszar roboczy nie daje żadnych efektów.
zblokowa-ne	1	nie ma	Nie ma nałożenia
		jest	Nie ma nałożenia
	2	nie ma	Nałożenie dla całego bloku
		jest	Nałożenie dla całego bloku. Obszar roboczy nie daje efektów.



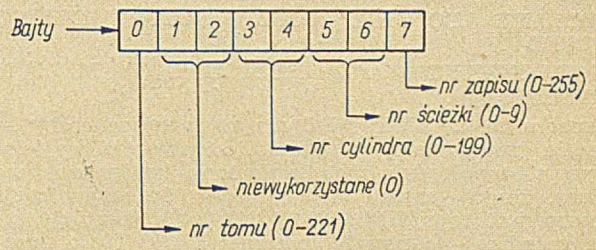
Rys. 5. Struktura zbiorów indeksowo-sekwencyjnych

Organizacja indeksowo-sekwencyjna wskazana jest w przypadkach, gdy przewiduje się zarówno dostęp bezpośredni, jak i sekwencyjny do tego samego zbioru, przy czym wiadomo, że jakaś pozycja zapisu może służyć jako klucz.

Zbiory indeksowo-sekwencyjne mogą być tworzone tylko na dyskach. Tabela II podaje pojemności ścieżek w pamięci dyskowej EC-5056 (pakiet dysków zawiera 200 ścieżek, a więc — przykładowo — na jednym pakiecie dysków zmieści się około 1200 zapisów o długości 500 znaków każdy, co dla wielu zastosowań jest liczbą zbyt małą).

Zbiory **zorganizowane losowo,** złożone z nieblokowanych zapisów długości stałej lub nieokreślonej mogą być zorganizowane w sposób umożliwiający bezpośredni dostęp. Pamięć dyskową dzieli się na obszary (czynność tę wykonuje standardowy podprogram formatyzacji CLRDSK), do których mogą być wprowadzone po jednym lub więcej zapisów. Dostęp do określonego zapisu w zbiorze możliwy jest dzięki temu, że występuje ścisła zależność pomiędzy kluczem zapisu, a jego adresem. Fizycznie, zapisy mogą być rozmieszczone na dysku w sposób losowy. Parametry zbioru należy podać instrukcją DTFDA.

Do programisty należy wybór najodpowiedniejszego algorytmu przyporządkowującego każdemu kluczowi fizyczny adres zapisu (tzw. randomizacja). Możliwy jest również dostęp bezpośredni według adresu fizycznego (identyfikatora) zapisu. Format adresu przedstawia rys. 6.



Rys. 6. Format adresu zapisu na dyskach

Zaletą losowej organizacji zbiorów jest to, że użytkownik ma zupełną swobodę w rozmieszczaniu zapisów. Czas dostępu jest stosunkowo mały (około 150 ms) i — w przybliżeniu — stały. Organizacja tego typu może być szczególnie przydatna w systemach typu „pytanie-odpowiedź". Istnieje możliwość przetwarzania zbioru losowego w sposób sekwencyjny.

Operowanie danymi w językach wyższego rzędu

Przedstawiony wyżej Logiczny SSWW można uznać za część składową języka ASSEMBLER. DOS JS umożliwia pracę również w językach programowania wyższego rzędu, z których każdy dysponuje własnym repertuarem środków operowania danymi. W każdym jednak przypadku, niezależnie od programisty, system posługuje się Fizycznym SSWW.

Język PL/1 w DOS JS zawiera wszystkie przedstawione wyżej metody organizacji zbiorów, używając przy tym znacznie prostszego zapisu. Możliwe są dwa sposoby przetwarzania zbiorów: zapisami (tryb RECORD) i strumieniem (tryb STREAM). W pierwszym typie przetwarzania, zapisy przenoszone są bez żadnych zmian z bufora urządzenia zewnętrznego do pamięci. Przy przetwarzaniu strumieniowym, zapisy znajdujące się w buforze nie są traktowane jako jedna całość, lecz przesyłane znakami lub grupami znaków z ewentualnym przekształceniem postaci danych. PL/1 pozwala organizować zbiory sekwencyjne, indeksowo-sekwencyjne i losowe. Pierwsze dwa rodzaje organizacji podobne są do uzyskanych za pomocą logicznego SSWW, natomiast zbiory losowe, umożliwiające dostęp bezpośredni mogą być dwóch rodzajów: REGIONAL (1) i REGIONAL (3). W pierwszym przypadku w każdym obszarze pamięci, wy-

Tabela II

Pojemność ścieżki pamięci dyskowej EC-5056

Maksymalna długość zapisu w bajtach		Liczba zapisów na ścieżce	Maksymalna długość zapisu w bajtach		Liczba zapisów na ścieżce
bez klucza	z kluczem		bez klucza	z kluczem	
3625	3605	1	161	142	16
1738	1719	2	148	129	17
1130	1110	3	136	117	18
829	810	4	126	107	19
650	630	5	117	97	20
530	511	6	107	88	21
446	426	7	100	81	22
382	363	8	93	74	23
333	314	9	87	68	24
293	274	10	81	62	25
261	242	11	76	57	26
234	215	12	71	52	27
212	192	13	66	47	28
192	173	14	62	43	29
175	156	15	58	39	30

różnionym swoim kluczem, może być wprowadzony jeden zapis, w drugim przypadku — na każdy taki obszar przypada więcej zapisów.
Na przykład opis zbioru losowego może być następujący:

DECLARE ABC FILE RECORD OUTPUT DIRECT KEYED
ENVIRONMENT (F (80) REGIONAL (1) MEDIUM (SYSO23,5056)).

Oznacza to, że zbiór ABC ma następujące cechy — będzie przetwarzany zapisami, jest zbiorem wyjściowym z bezpośrednią metodą dostępu, każdy zapis ma klucz, długość zapisu wynosi 80 bajtów, organizacja zbioru jest typu REGIONAL (1), zbiór znajduje się na urządzeniu o nazwie logicznej SYSO 23, a urządzeniem tym jest pamięć dyskowa typu EC-5056.

Odczytanie zapisu z tego zbioru odbywa się w wyniku działania instrukcji READ FILE ABC INTO (nazwa obszaru pamięci), KEY (klucz). Wprowadzanie zapisów do zbioru powoduje instrukcja WRITE, natomiast przy aktualizacji można użyć instrukcji REWRITE.

Język FORTRAN w DOS JS (BASIC FORTRAN) umożliwia sekwencyjny oraz bezpośredni dostęp do zbiorów. Podstawowymi instrukcjami w metodzie sekwencyjnej są READ i WRITE. Instrukcja READ ma format: READ a,b **lista,** co oznacza: czytaj dane ze zbioru z numerem **a** do obszarów pamięci, których nazwy wymienia lista. Dane przesyła się do pamięci zgodnie z formatem opisanym w instrukcji FORMAT, która ma etykietę **b.** Podobnie opisuje się instrukcję WRITE.

W celu zorganizowania na dysku zbioru z dostępem bezpośrednim, programista musi zadać w instrukcji DEFINE FILE charakterystykę zbioru (nr zbioru, liczba zapisów, maksymalna długość zapisu, nazwa zmiennej wskazującej nr zapisu). Zapisy w zbiorze są ponumerowane: pierwszy ma numer 1, drugi 2, itd. Programista chcący wyszukać i odczytać określony zapis musi podać w instrukcji READ (WRITE) nie tylko numer zbioru, ale również nr zapisu w zbiorze. Czas przetwarzania może być zmniejszony przez użycie instrukcji FIND, dzięki której w czasie przetwarzania bieżącego zapisu odbywa się wyszukiwanie następnego zapisu do przetwarzania.

Język RPG umożliwia stosowanie wszystkich trzech zasadniczych sposobów organizacji zbiorów (sekwencyjna, indeksowo-sekwencyjna, losowa). Zapisy mogą mieć stałą lub zmienną długość, mogą być blokowane lub nieblokowane. Zbiory opisuje się na specjalnych formularzach, co — być może — ułatwia pisanie programów, lecz pozbawia system elastyczności.

DOS JS zawiera również standardowe programy sortowania i łączenia zbiorów zapisanych na dyskach i taśmach.

Podsumowując, można stwierdzić, że największe możliwości daje Logiczny SSWW, a z języków wyższego rzędu jedynie PL/1 ma bogate i uniwersalne środki operowania zbiorami danych, pozwalające tworzyć również nietypowe struktury zbiorów (np. listujące, łańcuchowe), co może mieć istotne znaczenie przy organizowaniu banku danych. Opracowując swój system, użytkownik powinien uwzględnić specyfikę i możliwości wszystkich środków DOS, a następnie jego fragmenty — opracowane różnymi technikami i w różnych językach — łączyć za pomocą systemowego programu łączącego.

INSTYTUT MASZYN MATEMATYCZNYCH

Oddział Śląski

40-010 Katowice, ul. Warszawska 63

zatrudni:

- **inżynierów elektroników lub automatyków ze specjalnością maszyny matematyczne, miernictwo elektroniczne, teletransmisja, radio i telewizja, systemy kompleksowego sterowania i aparatura elektroniczna**
- **inżynierów mechaników — specjalność mechanika precyzyjna**
- **matematyków — numeryków i specjalistów w zakresie języków programowania**
- **ekonomistów — specjalizujących się w projektowaniu systemów epd**
- **konserwatorów i operatorów emc.**

Instytut ma możliwość zapewnienia wysokokwalifikowanym specjalistom mieszkań spółdzielczych

Zgłoszenia przyjmuje i informacji udziela Dział Kadr — Katowice, ul. Warszawska 63, tel. 380-54

WCT/1951/K/73

Metoda szacowania czasu realizacji prac programowych

W niniejszym opracowaniu wykorzystano materiały firmy ICL, a więc wszystkie podane normy czasowe i współczynniki wymagają sparwdzenia w naszych warunkach.
Wypracowanie odpowiednich norm czasowych i współczynników odpowiadających naszym metodom pracy wymaga kilkuletnich badań empirycznych. Ale też i poznanie przybliżonego czasu trwania prac programowych w celu opracowania odpowiedniego harmonogramu prac i sprecyzowania terminu zakończenia oprogramowania systemu jest bardzo potrzebne. Próby szacowania czasu pracy programisty mogą też prowadzić do wyeliminowania pewnych niedoskonałości organizacji prac programowych.

Opis metody

Aby określić czas realizacji prac programowych konieczne jest oszacowanie efektywnego czasu programowania i związanych z tym operacji. Szacowanie czasu realizacji prac programowych przebiegać będzie w trzech różnych etapach.
Pierwszy etap — to stadium projektowania założeń systemu. Na etapie tym szacujemy wstępnie czas prac programowych w celu określenia terminu zakończenia oprogramowania systemu. Już w tym okresie musimy korzystać z pewnych norm stanowiących podstawę, na której oparta będzie nasza kalkulacja. Normy takie podane są w tabelach I i II. Szacowanie to będzie się odbywać na podstawie niżej podanego wzoru i będzie wartością przybliżoną (brak dokładnie sprecyzowanych współczynników n i s).

$$\frac{n \times e \times s}{1000} = \text{pracochłonność w dniach}$$

gdzie n — liczba instrukcji programu
s — współczynnik trudności programu (tabela II)
e — pracochłonność jednostkowa programowania (tabela I)

Drugi etap pozwala na dokładniejsze, niż w etapie pierwszym, oszacowanie czasu prac programowych, gdyż w tym czasie dysponujemy już projektem technicznym i szczegółowymi założeniami programowymi wraz ze schematem blokowym (ogólnym), w związku z czym możemy dokładniej określić współczynniki n i s.
Oczywiste jest, że oszacowanie czasu prac programowych na etapie pierwszym może nieco różnić się od oszacowania na etapie drugim. Dla doświadczonego programisty, pracującego często nad oszacowaniem czasu, różnice te jednak nie powinny być duże.

Etap trzeci szacowania czasu polega na uwzględnieniu współczynnika wydajności programisty (tabela III).
Szacowanie w ten sposób czasu realizacji prac programowych ma na celu planowanie prac Działu Programowania, tzn. ustalenie terminów realizacji oprogramowania systemu i potrzebnych do tego środków (kadr programistów).
Wzór, według którego obliczamy czas programowania na tym etapie, ma postać:

$$\frac{n \times e}{1000} = s \times r = \text{pracochłonność w dniach}$$

gdzie r = współczynnik wydajności (tabela III).

Określenie języka programowania

Określenie (w przybliżeniu) liczby instrukcji, które będzie zawierał dany program jest możliwe po dokonaniu wyboru języka programowania dla danego problemu.
Dla przypadków, kiedy cały program (lub jego część) będzie pisany w innym języku niż ten, w którym dokonano wyceny — trzeba zastosować odpowiednie współczynniki przeliczeniowe liczby instrukcji.
W tabeli IV zawarte są współczynniki zależności pomiędzy językiem programowania, dla którego wykonano oszacowanie liczby instrukcji, a językiem programowania, w którym program będzie pisany.

Obliczając liczbę instrukcji „n" należy uwzględnić odpowiedni współczynnik z tabeli IV tak, aby ustalona liczba instrukcji „n" uwzględniała już odpowiednio zależność od języka programowania. Na przykład, jeżeli oszacowania liczby instrukcji dla programu wykonano z myślą o języku FULL COBOL, a program będzie pisany w języku COMPACT COBOL, liczbę instrukcji należy pomnożyć przez współczynnik 1,7.

Szacowanie liczby instrukcji

Dla doświadczonego programisty nie przedstawia większych trudności oszacowanie liczby instrukcji dla określonego programu. Faktem jest, że metoda szacowania liczby instrukcji będzie inna dla etapu pierwszego, a inna dla drugiego.
W pierwszym przypadku oszacowanie odbywać się będzie tylko na podstawie ogólnego schematu blokowego (tym samym wymagane jest duże doświadczenie pracownika w programowaniu i szacowaniu czasu prac programowych).
W drugim przypadku szacowanie odbywa się na podstawie dokładnych założeń programowych oraz schematów blokowych, co w dużym stopniu powinno ułatwić pracę.
Oczywiście inaczej będzie przebiegało szacowanie liczby instrukcji dla języka PLAN, a inaczej dla COBOLu. Dla PLANu określenie liczby instrukcji powinno wykluczać:

Tabela I
Pracochłonność jednostkowa programowania

Fazy prac programowych	Pracochłonność jednostkowa „e" przy różnych językach programowania					
	PLAN		COMPACT COBOL		FULL COBOL	
	liczba dni	%	liczba dni	%	liczba dni	%
1. Studiowanie założeń programowych	1	2,7	1,5	2,3	2	2,8
2. Układanie schematu blokowego + sprawdzenie	5	18,5	10,5	17,0	15	20,8
3. Kodowanie + sprawdzenie	12	32,5	20	32,3	20	27,8
4. Przygotowanie danych testujących	4	10,8	7	11,3	8	12,5
5. Kompilacja i przebieg próby programu	3	7,0	5	8,0	6	8,3
6. Testowanie	12	32,5	18	29,2	20	27,8
Razem dni	37	100	62	100	72	100
Razem godzin lub tygodni	300 godzin = 6 tygodni		500 godzin = 10 tygodni		580 godzin = 12 tygodni	

Tabela II

Współczynniki trudności programu

Typ programu	Współczynnik trudności (S)
PROSTY — zapis jednego typu danych na taśmę magnetyczną — drukowanie w jednym określonym formacie z jednego zbioru o bardzo prostej strukturze — wybieranie jednego typu danych z jednego zbioru — itp.	0,75—1
ŚREDNI — zapis kilku typów danych — drukowanie w kilku określonych formatach z jednego zbioru — wybieranie kilku typów danych z jednego zbioru — uaktualnienie jednego zbioru o prostej strukturze przez dwa do trzech typów aktualizatorów — itp.	1,00—1,25
SKOMPLIKOWANY — zapis kilku typów danych poszerzony o kontrolę tych danych — naliczanie w tabele (macierze) w pamięci operacyjnej 2—3 typów danych z jednego zbioru o prostej strukturze — uaktualnienie zbioru o skomplikowanej strukturze przez kilka typów aktualizatorów — kontrola kilku typów danych i druk błędnych zapisów z komentarzem błędu — itp.	1,25—1,50
BARDZO SKOMPLIKOWANY — naliczanie w tabele (macierze) w PAO kilku typów danych — uaktualnienie 2—3 zbiorów o skomplikowanej strukturze przez kilka typów aktualizatorów — itp.	1,50—2,0
OBLICZENIA W CZASIE RZECZYWISTYM	2,5—3,0

— wszystkie dyrektywy
— definicje pól (deklaracje stałych i zmiennych)
— komentarze

Wszystkie rozkazy typu MACRO powinny być liczone według liczby instrukcji napisanych, a nie wygenerowanych.

Dla COBOLu wszystkie słowa w części PROCEDURE DIVISION powinny być liczone jako instrukcje. Oszacowanie czasu na określenie i napisanie dyrektyw definicji obszarów roboczych, deklarację stałych i zmiennych oraz komentarzy, zarówno dla PLANu, jak i dla COBOLu powinno odbywać się na innych zasadach.

Procochłonność programu

Ustalając pracochłonność programu, która następnie jest podstawą do określenia zadań dla programisty, należy uwzględnić następujące czynniki:
— czas przestojów wynikających z nieprzewidzianych trudności i przeszkód
— niedoskonałość oszacowania czasu realizacji prac programowych w fazie projektowania. Wydaje się, że uwzględnienie powyższych czynników wymaga zwiększenia czasu prac programowych o 25%.

Tak więc wzór na obliczanie pracochłonności prac programowych otrzymuje postać

$$E = \frac{n \times e}{100} \times s + W$$

gdzie E — całkowity czas realizacji prac programowych
n — liczba instrukcji w programie
e — pracochłonność jednostkowa z tabeli I (w dniach)
s — współczynnik trudności programu — tabela II

$$W = \frac{n \times e}{1000} \times 0{,}25$$

zatem: $E = \frac{n \times e}{1000}(s + 0{,}25)$ dni

Współczynniki wydajności programistów

Oszacowanie czasu realizacji programu po uwzględnieniu osobowych cech programisty odbywa się w etapie trzecim. Zwiększenie (wydłużenie) czasu, wynikłe z oszacowania na tym etapie nie

Tabela III

Współczynniki wydajności programistów

Staż programisty w miesiącach	Współczynnik wydajności „r"
3—6	1,4
6—9	1,2
9—12	1,0
12—15	0,9
15—18	0,8
18—21	0,7
21—24	0,6
i więcej	0,5

może mieć wpływu na ostateczny termin realizacji prac programowych, gdyż jest to sprawa organizacji zespołu Działu Programowania. W tabeli I podano liczby określające pracochłonność w dniach dla poszczególnych faz prac programowych. Liczby zawartę w tabeli I dotyczą programu o średnim stopniu trudności (współczynnik trudności 1) i około 1000 instrukcji.

Tabela I zawiera pracochłonności jednostkowe dla progarmisty o 9—12 miesięcznym stażu (doświadczeniu). Przy ustalaniu wydajności należy koniecznie uwzględnić staż programisty, któremu powierzamy napisanie danego programu oraz jego indywidualne uzdolnienia.

W tabeli III zawarte są współczynniki wydajności w zależności od stażu programisty. Jeżeli nad programem pracować będzie dwóch lub więcej programistów, pracochłonność będzie odpowiednio mniejsza i tak: jeżeli pracuje dwóch progarimstów — pracochłonność mnożymy przez 2/3, jeżeli pracuje trzech programistów — pracochłonność mnożymy przez 1/2. Nie jest wskazane, aby nad jednym programem pracowało więcej niż trzech programistów.

Tak więc określając globalnie czas wykonania programu, bierzemy pod uwagę czas oszacowany na etapie drugim i mnożymy go przez współczynnik zaawansowania programisty (tabela II) oraz uwzględniamy liczbę programistów, którzy będą pracowali nad tym programem.

Testowania i próby

Testowanie programu rozdzielić można na następujące etapy:

● Kompilacja

Na tym etapie wykonuje się program wynikowy *(object program)* i eliminuje się błędy formalne. Maksymalna liczba prób nie powinna przekraczać 3.

● Testowanie segmentu lub całego programu na danych próbnych.

Na tym etapie odbywa się sprawdzenie i eliminacja błędów logicznych programu. Liczba takich testowań nie powinna przekroczyć 8 dla programu o tysiącu instrukcji bez względu na język programowania, w którym napisany jest program.

● Scalenie programu (łączenie segmentów)

Ten etap występuje tylko wtedy, kiedy program składa się z kilku segmentów i segmenty te są testowane oddzielnie.

Zakłada się, że liczba prób mających na celu połączenie wszystkich segmen-

Tabela IV

Współczynnik zależności pomiędzy językami programowania

		Użyty język programowania		
		PLAN	COMPACT COBOL	FULL COBOL
Język programowania, dla którego wykonano oszacowanie	PLAN	1,0	0,5	0,3
	COMPACT COBOL	2,0	1,0	0,6
	FULL COBOL	3,0	1,7	1,0

tów nie powinna przekroczyć trzech wejść na maszynę dla każdego segmentu. Wobec tego całkowitą maksymalną liczbę prób obliczamy, jak następuje: całkowita liczba prób =

3 (uzyskanie programu wynikowego) +

$$\frac{8 \cdot n \text{ istrukcji}}{1000} + 3 \text{ na łącznie (scalenie)}$$

każdego modułu
lub inaczej:

$$\text{całkowita liczba prób} = 3 + \frac{8 \times n}{1000} + 3(m-1)$$

gdzie: m — liczba segmentów
n — liczba instrukcji

Czas pracy komputera potrzebny do wytestowania programów

Czynniki mające wpływ na czas pracy komputera to:

— liczba instrukcji w programie
— system operacyjny
— typ komputera i jego konfiguracja
— język programowania
— rodzaj programu
— objętość danych do przetwarzania.

Określając czas pracy komputera należy uwzględnić wszystkie podane wyżej czynniki.

Metoda oszacowania czasu pracy komputera ze względu na swój zupełnie odrębny charakter powinna być przedmiotem oddzielnego opracowania (materiały TIMING firmy ICL).

* * *

Opracowanie niniejsze nie ma na celu podania obowiązującej recepty na określenie czasu realizacji prac programowych, a jest jedynie zasygnalizowaniem problemu. Opracowanie to należy również uważać jako zaproszenie do dyskusji, szczególnie nad tabelami zawierającymi współczynniki i normy czasowe. Bez powszechnego ustosunkowania się do zaproponowanych tu norm czasowych, metoda szacowania czasu realizacji prac programowych będzie metodą tylko teoretyczną i nieprzydatną do zastosowania w naszej praktyce krajowej.

Aleksander Kuźmin

Technika sieciowa w organizacji procesu projektowania systemów informatycznych

Terminowe i ekonomiczne przedstawienie jakiegoś organizmu organizacyjnego na ETO wymaga dokładnego planowania. Służąca temu celowi technika planowania sieciowego omawiana jest na przykładzie systemu EPD obrotu materiałowego.

Wdrożenie dużego systemu informatycznego w zasadzie jest przedsięwzięciem inwestycyjnym, nie tylko dlatego, że prace z tym związane rozciągnięte są często na lata i ponoszone są duże nakłady finansowe, lecz również dlatego, że wdrożony system będzie musiał służyć przez wiele lat. Dziś już nie do pomyślenia jest aby realizacja jakiejś dużej inwestycji prowadzona była bez planowania. Wychodząc z tego zakłada się, że techniki planowania sieciowego mogą stanowić również optymalne narzędzie dla prowadzenia prac projektowych.

FUNKCJE PLANU SIECIOWEGO

Jak wiadomo plan sieciowy jest modelem do rozwiązywania kompleksowych zadań technicznych. Uwidacznia on logiczne powiązania poszczególnych działalności w ich przebiegu czasowym. Opracowanie planu sieciowego, po pierwsze zmusza do dokładnej analizy wszystkich koniecznych działalności (czynności) analizy czasu ich trwania oraz ich koordynacji. Po drugie — stanowi dobre narzędzie planowania i kontroli. Funkcje te idealnie odpowiadają wymaganiom, które muszą być spełnione podczas projektowania systemów informatycznych.

OPRACOWANIE PLANU SIECIOWEGO

Analiza projektu

W każdym projekcie informatycznym duża ilość pojedyńczych działań musi być przedstawiona w logicznym powiązaniu. Planowany projekt systemu musi być przeanalizowany ze względu na poszczególne działania. W zależności od indywidualnych cech głównego projektanta i od wielkości rozwiązywanego zadania zależny jest sotpień podziału prac projektowych na poszczególne działania.

W przypadku, gdy przeprowadzany jest podział do najniższego szczebla, często zaskakuje fakt, jak z wielu pojedyńczych kroków pracy składa się kompleksowe zadanie.

Podczas analizy poszczególnych działalności posługujemy się definicją systemu, podsystemu, jednostki funkcjonalnej, modułu [1]).

Celem uświadomienia sobie tego procesu najlepiej posłużyć się przykładem z problematyki systemu obrotu materiałowego. Nadmienić należy jednak — że chodzi tu o prezentację jedynie sposobu podejścia, dlatego też nie będzie przedstawionoy pełen zakres działalności występujący w danym projekcie.

W pierwszym etapie analizy definiowane będą tylko projekty częściowe (podsystemy).

Jako projekt częściowy zdefiniowany jest tu więc określony obszar problemu, który w dużym stopniu lub w pełni będzie mógł być rozwiązywany niezależnie w ramach całego projektu. W przykładzie naszym mogą to być:
sprzedaż wyrobów
zakup materiałów
planowanie środków transportowych
obliczanie opłat za transport.

Podczas już tego ogólnego podziału całego projektu widocznym jest, które części projektu będą mogły być realizowane czasowo jeden za drugim, względnie równolegle. Oczywiście decyzje te w większym stopniu zależne są od podejścia projektanta aniżeli wypływające z przesłanek rzeczowych.

W drugim kroku podsystemy te podzielone będą na mniejsze jednostki. Na przykład: jeżeli wybierzemy z naszego przykładu „Obliczanie opłat transportowych", można wyróżnić tu następujące jednostki funkcjonalne:

Obliczanie przewidywanych opłat za transport

Obliczanie aktualnych opłat za transport

Kontrola opłat za transport
Podział kosztów transportu
Statystyka

Te jednostki funkcjonalne mogą być podzielone na dalsze części. Na przykład jednostkę funkcjonalną „Obliczanie opłat przewidywanego transportu" można podzielić na:

Opłaty kolejowe
Opłaty lotnicze
Opłaty drogami wodnymi

Opłaty samochodowe

Każda z tych części w pojedyńczych przypadkach może być jednak tak obszerna, że może być rozpatrywana jeszcze jako samodzielny projekt. W tym wypadku przewidywać można dalszy podział według obszarów geograficznych (opłaty w kraju lub zagranicą). Załóżmy jednak, że znajdujemy się na najniższym stopniu analizy i w tym wypadku należy zdefiniować różne działania (wymogi) konieczne do wykonania omawianego projektu. Podczas opracowywania projektu częściowego (modułu lub jednostki funkcjonalnej) można wyróżnić m. in. następujące czynności:

Analiza stanu istniejącego
Opracowanie koncepcji systemu
Planowanie kosztów
Zatwierdzanie do realizacji
Dostarczanie urządzeń

[1]) „Metodyka projektowa" Wyd. OBRI. Warszawa 1974 r.

Definiowanie danych wejściowych
Definiowanie wydawnictw wynikowych
Opracowanie tabulogramów
Analiza możliwości systemu operacyjnego
Definiowanie zbiorów głównych (kartoteki stałe)
Definiowanie zbiorów pomocniczych
Definiowanie tabel
Definiowanie systemu indeksowania (klucze)
Definiowanie sposobu dostępu do zbiorów
Przekazywanie danych dla innych programów
Programowanie i uruchomienie programów
Aktualizowanie danych
Szkolenie
Konserwacja programów

Te poszczególne działania rozpatrywane mogą być jako makrodziałania gdyż na ogół składają one się z większej lub mniejszej ilości pojedyńczych działań. Pomimo to, z tak zdefiniowanymi poszczególnymi czynnościami, można już dalej pracować. Decyzja w tym zakresie zależy w dużej mierze od indywidualnych cech głównego projektanta. Ze względu jednak na konieczność szacowania czasu wydaje się celowym przeprowadzenie podziału na pojedyńcze działania. Omówmy ten krok na przykładzie. Spróbujmy punkt „Zbiór pomocniczy" podzielić na dalsze czynności. Wiadomym jest z praktyki, że na ogół występuje większa liczba takich zbiorów. Jest więc możliwym i celowym rozpatrywanie łączne poszczególnych działań dla tych zbiorów. Ponieważ jednak przy różnych zbiorach pracują różne grupy ludzkie, a również występują przesunięcia czasowe w pracy nad różnymi zbiorami, jest celowym opracowanie dla każdego zbioru własnego zakresu poszczególnych działalności, który może wyglądać następująco:

Analiza potrzebnych danych
Usystematyzowanie danych
Ustalenie budowy zapisu
Opracowanie formularzy dla danych
Skatalogowanie danych
Wypełnienie formularzy danymi
Przeniesienie zawartości formularzy na karty dziurkowane
Analiza systemu i programowanie dotyczące rejestracji danych
Budowa zbioru — przebieg zakładania i wydruk kontrolny zbioru
Sprawdzenie zbioru (wykrycie błędów) na komputerze
Wprowadzenie poprawek
Przebieg komputerowy z poprawkami
Organizacyjne ustalenie obsługi konserwacyjnej zbioru
Ustalenie sposobu przeprowadzenia konserwacji

Każdy moduł, jednostka funkcjonalna lub system może być podzielony systematycznie. Przy tym liczba pojedyńczych czynności — jak już wspomniano — dla całego projektu, może być dość duża. Jeżeli chcemy liczbę tę zmniejszyć to również określone makro-działania mogą być przedstawione łącznie. W ten sposób może być tworzone dla wszystkich jednostek funkcjonalnych jedynie jedno działanie „analiza stanu istniejącego" lub „programowanie". Ponieważ jednak generalnie zakładamy, że właściwym środkiem dla opracowania planu sieciowego jest komputer, należy dążyć do podziału szczegółowego. Jest bardzo prawdopodobne, że zarówno makro-działania jak i działania pojedyńcze w każdym projekcie będą wielokrotnie powtarzać się.

Omawiane w przykładzie makro-działania mogą wystąpić w 4 naszych modułach. Oznacza to, że na ogół biorąc opracowany wykaz działań będzie mało się różnić z projektu na projekt. W tym wypadku środkiem pomocniczym może być raz opracowana lista działań którą można wykorzystywać podczas każdego projektowania.

LOGICZNE UPORZĄDKOWANIE DZIAŁAŃ

Po ustaleniu poszczególnych działań, muszą one być uporządkowane w taką kolejność w jakiej będą dalej opracowywane. W tym przypadku pierwszym krokiem jest katalogowanie. Dla każdej działalności przyporządkowany jest identyfikator. Najczęściej jego struktura zależy od głębokości przeprowadzonego podziału i od standardowego programu sieciowego istniejącego w danym systemie komputerowym.

Wydaje się celowym przyporządkowywanie poszczególnym częściom systemu jako identyfikatorów liter, np.

A = Obliczanie opłat aktualnych
P = Opłaty przewidywane
K = Opłaty kolejowe itp.

Następnie poszczególne działania mogą być numerowane w sposób usystematyzowany lub nieusystematyzowany. W ten sposób np. działanie „Opracowanie formularzy dla danych" może mieć symbol APK-18. W tym wypadku jest szczególnie ważne przydzielenie jednoznacznych identyfikatorów dla poszczególnych działań.

Dla każdej czynności musi być ustalone miejsce logiczne w takiej kolejności w jakiej ono będzie przetwarzane. Przy tym muszą być wyodrębnione czynności, które są przeprowadzane w czasie kolejno jedna za drugą oraz przebiegające równolegle. Przebieg czynności jednej za drugą oznacza logiczną kolejność. Na przykład wydruk tabulogramu może nastąpić po jego uprzednim zaprojektowaniu. Testowanie jakiegokolwiek programu zależne jest od uprzedniej czynności kodowania itp.

Analiza związków czasowych dotyczy w zasadzie poszczególnych działań występujących w jednostce funkcjonalnej lub module.

Jak już nadmieniliśmy występują, także czynności równoległe. Są one w zasadzie określane przez to, że muszą być zakończone w określonym czasie, aby realizacja całego przedsięwzięcia nie była narażana na niebezpieczeństwo nieterminowego wykonania. Związki te należy zachować w opracowanym katalogu, w którym koniec jednego działania powiązany jest z działaniem drugim.

Przy ustalaniu logicznej struktury głównie chodzi o optymalny model, który może być zmieniony przez inne czynniki.

WYKORZYSTANIE PLANU SIECIOWEGO

Pierwszym czynnikiem mającym wpływ na logiczną strukturę poszczególnych działań jest czynnik pracy. Cały zakres działań nie może być napewno zrealizowany przez jedną osobę, ale również nie przez tak wiele osób, aby wszystkie teoretycznie możliwe równoległe działania, faktycznie mogły być przeprowadzone równolegle.

Dla przeprowadzenia prac projektowych i wdrożeniowych na ogół stoi do dyspozycji ograniczona liczba pracowników zarówno z komórki projektowania oraz ośrodka obliczeniowego jak i innych działów przedsiębiorstwa. Najczęściej w ramach działów poszczególne osoby lub grupy będą wyłączone od tych prac. Prócz tego dla wykonania centralnych prac powoływana jest najczęściej grupa projektantów składająca się z wszystkich działów, których dotyczy opracowywany system.

Teraz w następnych krokach dla poszczególnych działań przydzielone będą osoby względnie grupy osób, których zadaniem jest wykonanie projektu. Decyzje przyporządkowania powinny opierać się na przesłankach rzeczowych a więc temu kto najlepiej może wykonać projekt.

PRZYPORZĄDKOWANIE CZASÓW

Po przyporządkowaniu na każde działanie (czynności) osób będących do dyspozycji może być podany czas trwania tych działań. Czasy te są na ogół wielkościami szacunkowymi a wystarczającą dokładnością jest podanie ich w dniach lub tygodniach.

Należy jednak zwracać uwagę, aby poszczególne szacunki czasów szczególnie przy dużych projektach nie były zaniżone.

Należy zwrócić uwagę, że działają wydłużająco czasy spowodowane chorobami, urlopami, które nie są rozkładane na poszczególne działania, jak i nie przewidywane opóźnienia natury osobowej lub rzeczowej.

ZESTAWIENIE PLANU SIECIOWEGO

Bez wątpienia jest możliwym zestawienie planu sieciowego ręcznie. Ponieważ jednak w złożonych projektach nakład pracy jest zbyt duży, często przekraczający możliwości realizacyjne, dlatego należy wykorzystywać do tych czynności komputer.

Większość producentów komputerów oferuje standardowe pakiety programów z zakresu metod sieciowych, nie zachodzi więc potrzeba opracowywania własnych programów. Oczywiście wymagany jest znaczny nakład pracy dla przygotowania danych wejściowych.

Przy pomocy standardowych programów na podstawie podanych zależności, każde działanie będzie zaopatrzone w termin rozpoczęcia i zakończenia. Równocześnie podawana jest ścieżka krytyczna i wyliczony planowy termin zakończenia. Dla wszystkich nie będących na ścieżce krytycznej działań obliczane są zapasy czasów, które podają w jakim odstępie czasu może być przeprowadzona praca.

Ponieważ w pierwszym przebiegu przetwarzania, kolejność poszczególnych działań była definiowana na podstawie logicznych zależności gromadzonych historycznie (czasowo), istnieje możliwość iż mimo występujących zapasów czasów za dużo równoległych działań przypada na jakąś osobę względnie grupę osób. Dlatego musi być przeprowadzona optymalizacja działań ze względu na czynnik pracy. Stąd pewne zależności muszą być częściowo zdefiniowane na nowo. Poza tym istnieje również możliwość, że wyliczony termin zakończenia prac jest zbyt odległy.

W tym przypadku istnieje możliwość dodania osób do działań znajdujących się na ścieżce krytycznej i skrócenia planowanego terminu zakończenia tych prac. Zmiana wielkości czynnika pracy może spowodować, że w drugim, trzecim lub dalszych przebiegach przetwarzania, inne działania będą bardziej krytyczne aniżeli poprzednio. W pełni ukazują się tutaj korzyści stosowania elektronicznego przetwarzania danych, gdyż wprowadzenie zmian wymaga stosunkowo małych nakładów pracy natomiast cały ciężar przeliczeń przejmie system liczący.

WYKORZYSTANIE PLANU SIECIOWEGO

Poprzez jasne ustalenie terminów w planie sieciowym wszystkie działania zostają wyraźnie ustalone i wydrukowane. Zależy teraz od głównego projektanta w jakim stopniu te informacje będą wykorzystywane, aby np. na określony punkt czasowy przydzielić do wykonania określone działanie projektantowi lub grupie projektowej.

Większą przejrzystość wykonywanej pracy można osiągnąć przez wykreślanie diagramów słupkowych. Kolorowe oznaczanie różnych osób wzgl. zespołów, daje wartościową pomoc dla kierowania pracami projektowymi.

Z działaniem dyspozycyjnym również ściśle łączą się funkcje kontrolne. Jeżeli główny projektant przeprowadzi rozdział pracy, będzie on w sposób konsekwentny informowany o jej wykonywaniu, już chociażby dlatego, że założeniem tej metody jest posiadanie wyników przed podejmowaniem nowych działań. Główny projektant może również w zasadzie ograniczyć się do kontroli terminów. W tym przypadku korzystnym jest posiadanie szczegółowego podziału czynności.

Opracowanie całego projektu na czasowo wiele krótkich działań daje możliwość rozpoznawania wcześnie wszelkich opóźnień i podejmowanie odpowiednich przedsięwzięć eliminujących się.

Może również zdarzyć się konieczność wprowadzenia nowych działań do sieci. Są one włączane do całego planu bez żadnych przeszkód. W przypadku jednak potrzeby wprowadzania poprawek lub przesunięć czasowych, zalecane jest opracowanie nowego planu sieciowego. Ponieważ opracowanie planu sieciowego związane jest z dość dużym nakładem pracy, wykorzystanie tej metody jest dopiero wtedy korzystne, gdy projektowany system ma określoną wielkość.

Antoni Bielak

Standardy dokumentacji projektowej, programowej, i eksploatacyjnej OBRI

Z inicjatywy Zjednoczenia Informatyki podjęto w Ośrodku Badawczo-Rozwojowym Informatyki (OBRI) prace nad ujednoliceniem:

— głównych zasad konstrukcji oraz postępowania przy projektowaniu i wdrażaniu systemów informatycznych, oraz
— standardów dokumentacji projektowej, programowej i eksploatacyjnej systemów informatycznych.

Prace te zostały pomyślnie zakończone m.in. dzięki aktywnej współpracy najbardziej doświadczonych projektantów i programistów wielu ZETO. Opracowane zostały:

1. Metodyka projektowania systemów informatycznych (w zakresie etapowania prac oraz składu, treści i form dokumentacji systemu); część I [1]).

2. Metodyka projektowania systemów informatycznych (wzory formularzy — standardów dokumentacji projektowo-programowej i eksploatacyjnej oraz ich opis); część II.

Zarządzenie Naczelnego Dyrektora Zjednoczenia Informatyki obliguje wszystkie jednostki organizacyjne do stosowania jednakowych zasad (wg ujętych w w/w dwóch opracowaniach) projektowania i dokumentowania systemów, rozpoczynanych po 1.07. br.

Wyżej użyłem określenia: „Prace te zostały pomyślnie zakończone...". Rozumieć należy to jako zakończenie pewnego etapu prac. Metodyka projektowania systemów informatycznych (bez względu na zakres) jest zawsze sprawą otwartą. Można chyba stwierdzić, że metodyka jest w dużej mierze funkcją kwalifikacji kadr informatycznych, stanu sprzętu komputerowego i jego oprogramowania, a ponieważ czynniki te rozwijają się dynamicznie, więc metodyka projektowania musi być okresowo aktualizowana.

Poniżej przedstawiam krótką charakterystykę formularzy — standardów dokumentacji projektowej, programowej i eksploatacyjnej.

Głównym celem unifikacyjnej działalności w zakresie tych standardów (przynajmniej dla Zjednoczenia Informatyki) jest to, aby opracowanie jednego ośrodka obliczeniowego mogło być bez przeróbek dokumentacji wykorzystywane przez inny dowolny ośrodek, w związku z czym główny nacisk musi być położony na standardy dokumentacji projektu technicznego, programów i eksploatacji.

Formularze — standardy można podzielić na trzy grupy:

1. Formularze — standardy ogólnego przeznaczenia.
Standardy te są przeznaczone dla dowolnej dokumentacji systemu informatycznego. Niektóre z nich są bez nazwy, ponieważ ich przeznaczenie może być różne.

2. Formularze — standardy przeznaczone do udokumentowania analizy istniejącego systemu przetwarzania danych. W tej grupie przedstawia się dwie wersje:
a) dla bardzo szczegółowego opisu istniejącego stanu i często w warunkach, gdy w danym obiekcie działa system informatyczny, a chodzi o jego poważne modyfikacje (zbliżone do najnowszych standardów ICL),
b) dla przypadków innych niż w wersji „a", przy czym niektóre mogą być wykorzystane również w wersji „a".

3. Formularze — standardy przeznaczone do opisu projektu technicznego, programów i dokumentów eksploatacyjnych.

Przedstawione tu są najważniejsze standardy, a głównym ich celem jest usprawnienie współpracy między poszczególnymi ośrodkami obliczeniowymi. Większość standardów tej grupy jest niezależna od typu komputera, na który się projektuje (programuje) i na którym się eksploatuje system informatyczny. Niektóre standardy ukierunkowane są na komputery ODRA serii 1300 i do nich zbliżone (ICL, MIŃSK 32, ...), a niektóre inne uwzględniają specyfikę i wymagania Komputerów Jednolitego Systemu i IBM (360, 370). Należy tu zwrócić uwagę na fakt przedstawienia odrębnych standardów dla ujęcia organizacji kartotek (plików, zbiorów) na dyskach ICL i JS (IBM); alternatywą tego mogło być opracowanie jednego ogólnego standardu, z pozostawieniem wolnych obszarów dla dopisania pewnych charakterystyk, specyficznych dla konkretnych rodzajów dysków magnetycznych, ale po dyskusji z przedstawicielami niektórych środowisk informatycznych alternatywę tę zarzucono.

Standaryzacji wymaga jeszcze szereg innych dokumetów. Wymagana będzie analiza doświadczeń, wynikających z wdrożenia i praktycznego wykorzystania tej wersji standardów w pewnym określonym przedziale czasowym.

Zapraszamy do współpracy wszystkich zainteresowanych omówionym tematem.

Henryk Zygier

[1]) Zarys tej części oraz model rozwiązania był przedmiotem narady z udziałem m.in. przedstawicieli wszystkich resortowych centrów informatyki.

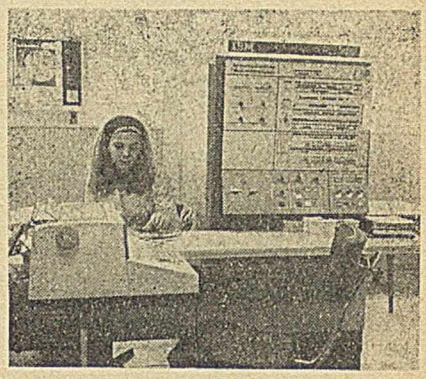
Starachowicki sposób na amortyzację komputera

Choć przedmiotem reportażu będzie komputer, wpierw parę słów o terenie na jakim się znajduje. Starachowice — miasteczko zalegające skraj przedwojennego COP-u, dziś 40 tysięczne, silnie uprzemysłowione — parę sporych zakładów a wśród nich kolos na skalę europejską: Fabryka Samochodów Ciężarowych. Stąd wywodzi się spora rodzina popularnych „Starów", samochodów, które dobrze od początku służą krajowej gospodarce, zaś eksportowane do szeregu krajów zyskały tam wysoką ocenę.

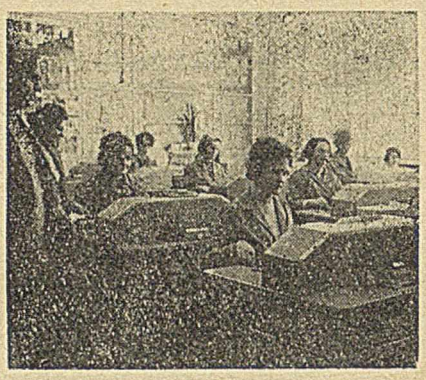
Historia powstania Fabryki jest długa i charakterystyczna dla okresu odbudowy polskiego przemysłu. Zaczyna się właściwie w roku 1948, gwoli sprawiedliwości należy jednak nadmienić, że nie powstała ona na pustyni. W okresie międzywojennym zbudowano tu duże zakłady przemysłu maszynowego. Tradycji więc nie brakło, ale tylko tradycji, bowiem co do maszyn i urządzeń — to te nieomal w komplecie wycofujacy się Niemcy wywieźli w głąb Rzeszy. Tak więc Fabryka musiała powstać niejako na nowo.

Obecnie jest to nowoczesne przedsiębiorstwo wielozakładowe, skupiające 3 zakłady zlokalizowane w Starachowicach i Zakłady w Końskich, Skarżysku, Iłży i Lipsku. Fabryka zatrudnia — bagatela — 27 tys. pracowników. Ciekawostka: 20 tys. pracuje w samych Starachowicach — miasteczku 40 tysięcznym co oznacza, że co drugi statystyczny obywatel pracuje w FSC. Jest to oczywisty absurd, gdy uwzględni się proporcję kategorii mieszkańców w wieku produkcyjnym i nie produkcyjnym. Wyjaśnienie jest takie, że Fabryka zatrudnia pracowników pochodzących spod miasta i to z takim skutkiem, że — jak słyszałem — w promieniu 60 kilometrów trudno znaleźć człowieka bez pracy.

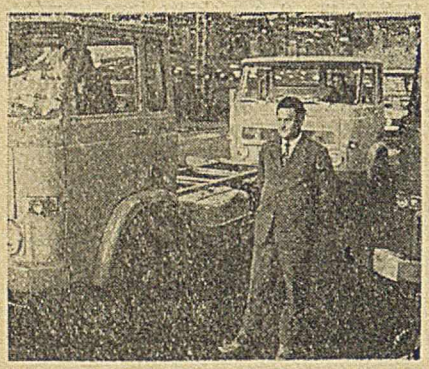
Piszę tylko pozornie nie na temat. Także sytuacja demograficzna Starachowic podyktowała Fabryce konieczność instalacji komputera. Ponieważ trudno o żywych ludzi — trzeba było zastąpić ich maszyną. Nie była to oczywiście jedyna przyczyna. Inne dyktowała organizacja i technologia produkcji.



Komputer IBM 360/40.
Maszyna wkrótce zostanie przeniesiona do nowego pomieszczenia.



Sala przygotowywania danych. Podstawowym nośnikiem w FSC są karty, których zużywa się pół miliona miesięcznie. Dziurkarki IBM, ARITMA i SOEMTRON w ilości łącznej 30 sztuk najlepiej obsługują widoczne na foto: A. Witkos, M. Losch, B. Zięba, B. Kozłowska, T. Lipa, B. Mrozowska, D. Każdub — pracowniczki działu kierowanego przez M. Zara.



Mgr inż. Tadeusz Banach — dyrektor ekonomiczny FSC w Starachowicach. Nim został Dyrektorem był informatykiem. Oczywiście, że był gorącym rzecznikiem wprowadzenia ETO. Twierdzi, że pozbawiony komputera — Zakład pogrążyłby się w totalnym chaosie.
W tle hala montażu STARÓW, Starachowice produkują ich dziesiątki tysięcy rocznie dla kraju i zagranicy.

Współcześnie produkowane „Stary" cechuje 8-krotny stopień montażu, 8 tysięczny indeks elementów składających się na produkt ostateczny i 170 tys. tzw. „detalo-operacji". Było więc jasne, że dla komputera trudno o wdzięczniejszy teren do weryfikacji jego możliwości. Zwłaszcza, że elektroniczna technika obliczeniowa weszła już wcześniej do systemu zarządzania Fabryką, a mianowicie już w 1965 roku. Był to okres przejścia na produkcję wielkoseryjną; dla potrzeb planowania i zarządzania Starachowice korzystały już z usług ZOWAR-u. Dużo wcześniej bo już w 1954 roku proste obliczenia (zwłaszcza płace) prowadzono na zestawach MLA-SAM a nieco później na ARITMACH.

Rosnąca ilość zadań wyznaczała konieczność uzyskania własnego komputera, zwłaszcza że czas pracy maszyny jaki mógł udostępnić ZOWAR (80—120 godzin miesięcznie) począł być niewystarczający.

W marcu 1972 r. Ośrodek Przetwarzania Informacji FSC kosztem 1 mln dolarów otrzymuje komputer IBM-360 o następującej konfiguracji: jednostka centralna o pamięci operacyjnej 128 KB, pamięć zewnętrzna: 6 jednostek dyskowych o pojemności 29 mln bajtów każda oraz 4 jednostki taśmowe, 1 czytnik kart 90- i 1 80-kolumnowych, drukarka wierszowa o szybkości 1300 wierszy/min. oraz 1 końcówka ekranowa.

W trzy miesiące po zainstalowaniu komputer osiąga 250 godzin pracy miesięcznie, obecnie liczba ta uległa podwojeniu, przy czym warto nadmienić, że dotychczas nie zanotowano żadnej poważniejszej awarii.

Należy przyznać, że jak rzadko gdzie, pole do wykorzystania komputera było przygotowane znakomicie. Nie stał się więc on kłopotliwym podarkiem, lecz konsekwentnym dopełnieniem z dawna nakreślonej koncepcji. Jeszcze w okresie korzystania z uczynności ZOWAR-u nakreślono koncepcję systemu. W 1970 roku opracowano założenia projektu systemu EPD. Baza danych zawiera informacje o strukturze wyrobów, o częściach i

materiałach, technologii, środkach produkcji i o personelu Fabryki. Dane te są podstawą do funkcjonowania szeregu podsystemów. Dotychczas wdrożono następujące podsystemy:

● Podsystem „Techniczne Przygotowanie Produkcji" obejmujący: emisję planów dla narzędziowni, plany dostaw dla zaopatrzenia, zestawienie części zespołów, przygotowanych do wykonania serii próbnej, plany opracowania dokumentacji i oprzyrządowania, sprawozdania z przebiegu realizacji przygotowania produkcji, wykonawstwa części i zespołów oraz zaopatrzenia materiałowego dla serii próbnej.

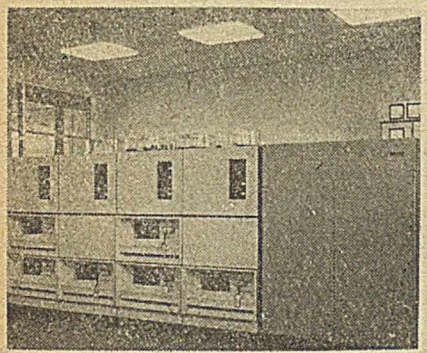
● Podsystem „Produkcja podstawowa" obejmujący: prognostyczne plany wieloletnie przedsiębiorstwa, plany: roczne, kwartalne, miesięczne i dekadowe w/g ilości asortymentu towaru i wartości, kontrolę realizacji planów, ewidencję wykonanej produkcji oraz sprawozdawczość miesięczną, kwartalną i roczną.

● Podsystem „Gospodarka materiałowa" obejmujący: ramowe plany kooperacji i zaopatrzenia materiałowego, kontrolę realizacji dostaw, ewidencję obrotów i stanu magazynów.

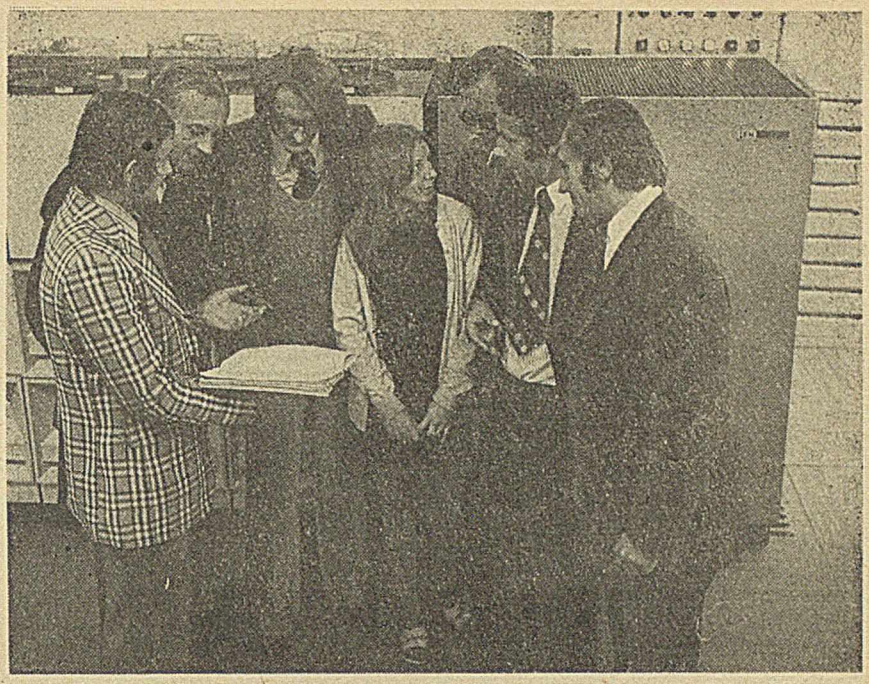
● Podsystem „Zatrudnienie i płace" obejmujący: wieloletnie plany (przybliżone) pracochłonności, produkcji i zatrudnienia dla przedsiębiorstwa (także roczne i kwartalne), kontrolę pracochłonności, zatrudnienia i funduszu płac oraz obliczanie zarobków i wydajność pracy.

● Podsystem „Koszty wytwarzania" obejmujący: wyliczenia i aktualizację norm, kosztów, wyceny robót w toku, wyceny wykonanej produkcji, rzeczywiste koszty wytwarzania.

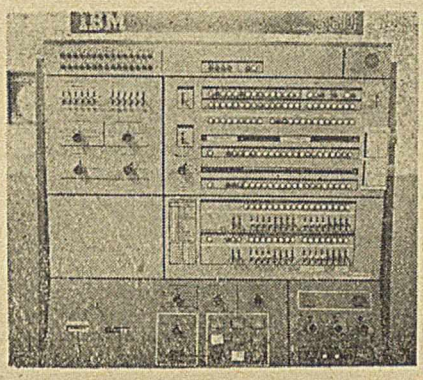
● Podsystem „Gospodarka maszynami" obejmujący: bilansowanie zdolności produkcyjnej stanowisk, zestawienie wykonanych norm na żądanych stanowiskach roboczych, zestawienie godzin pracy i przestojów maszyn i urządzeń, ewidencję i amortyzację środków trwałych.



Jednostki dyskowej pamięci zewnętrznej. Na 6 jednostek pamięci Ośrodek zabezpieczył 40 dysków o pojemności 29 mln bajtów każdy.



Zespół projektantów — programistów, reprezentują tu mgr K. Mińczyk, mgr S. Jeżewski, inż. M. Kowalczyk, mgr B. Jaciubek, inż. S. Walkiewicz, E. Lis i A. Kojek, którzy położyli największe zasługi przy tworzeniu podsystemów EPD. Ogólny stan zatrudnienia w Ośrodku oscyluje koło 160 osób. W większości kadra pochodząca ze Starachowic, wykształcona w uczelniach całej Polski. Na ogół odporni na pokusy, które stwarzają wielkie miasta, ale fluktuacja jest: w ostatnich latach ośrodek opuściło 17 osób.



Pulpit operatorski komputera IBM.

● Podsystem „Informacja kierownictwa" obejmujący: zapis dziennych transakcji z wykonania produkcji i dostaw kooperacyjnych, meldunki okresowe dla Głównego Dyspozytora z przebiegu produkcji, informacja o realizacji produkcji i sprzedaży (przekazywana na końcówkę ekranową).

Wymienione podsystemy podlegają systematycznej rozbudowie, zaś system jest stale wzbogacany o nowe podsystemy — np. latem ub.r. wdrożono podsystemy:

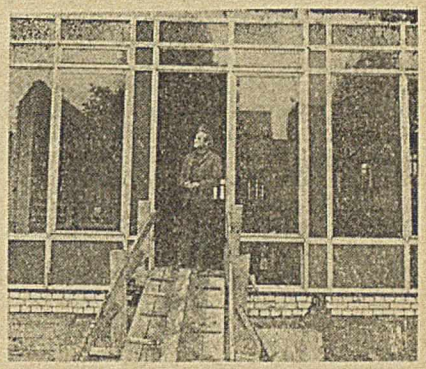
● gospodarka narzędziowa

● gospodarka wyrobami gotowymi

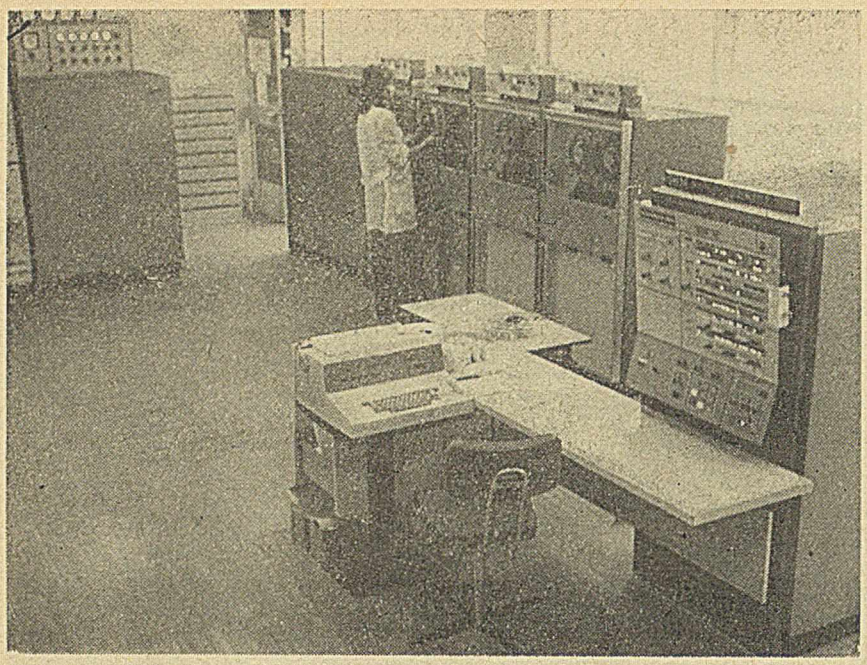
● inwestycje

Oprogramowanie systemu opracowywane jest w 65% własnymi siłami (dotychczas opracowano ca 800 programów); w 25 procentach korzysta się z pakietów zastosowań IBM zaś w 10% z programów ogólnego użytku IBM.

Wydaje się że dalsza rozbudowa systemu może napotkać na trudności płynące z dość ubogiej konfiguracji komputera. Efektywność maszyny mogłaby znacznie wzrosnąć przy przejściu na pracę wieloprogramową w systemie operacyjnym DOS (lub OS). W tym celu niezbędna jest rozbudowa pamięci operacyjnej o dalsze 128 KB



Mgr inż. Stanisław Pluciński — Kierownik Ośrodka Przetwarzania Informacji z utęsknieniem śledzi postęp prac w budynku Ośrodka. Z jego ukończeniem wiąże się nadzieje na radykalną poprawę warunków pracy. Sala komputera będzie miała powierzchnię 140 m². „Oby tylko" — martwi się szef ośrodka — „nie dokwaterowano nam zazdrośników". Na tle przedwojennej zabudowy terenu fabryki, pawilon Ośrodka stanowić będzie enklawę nowoczesności.



Do taśmowych jednostek pamięci zewnętrznej Ośrodek stosuje taśmy MEMOREX.

Czas po temu najwyższy albowiem warunki do pracy są bardziej niż kiepskie i nie korespondujące z wysoką klasą maszyny, która to maszyna współkreuje FSC na zakład o standardzie światowym. Zresztą do celowości instalowania komputera nie trzeba było tu nikogo długo przekonywać.

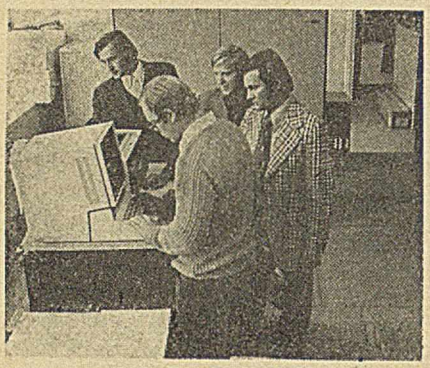
Efekty z zastosowania komputera są ewidentne i wymierne. Przykładowo — przejście na elektroniczny system przetwarzania rocznie przysparza 45 mln złotych zysku w wyniku następujących efektów organizacyjnych:

- zmniejszenie zapasu części gotowych
- zmniejszenie zapasów dla robót w toku
- zwiększenie współczynnika wykorzystania maszyn
- poprawę rytmiczności produkcji
- zmniejszenie przestojów
- zmniejszenie ilości godzin nadliczbowych
- kompresja etatów w służbach planistyczno-sprawozdawczych

Są jeszcze i inne efekty, które trudno wprawdzie przeliczyć na złotówki, ale są one aż nadto widoczne — głównie chodzi tu o szybką i dokładną informację.

Nie zdziwi zatem nikogo informacja, że komputer IBM osiągnął już prawie całkowitą amortyzację poniesionych nakładów inwestycyjnych po zaledwie 2 i pół rocznej eksploatacji.

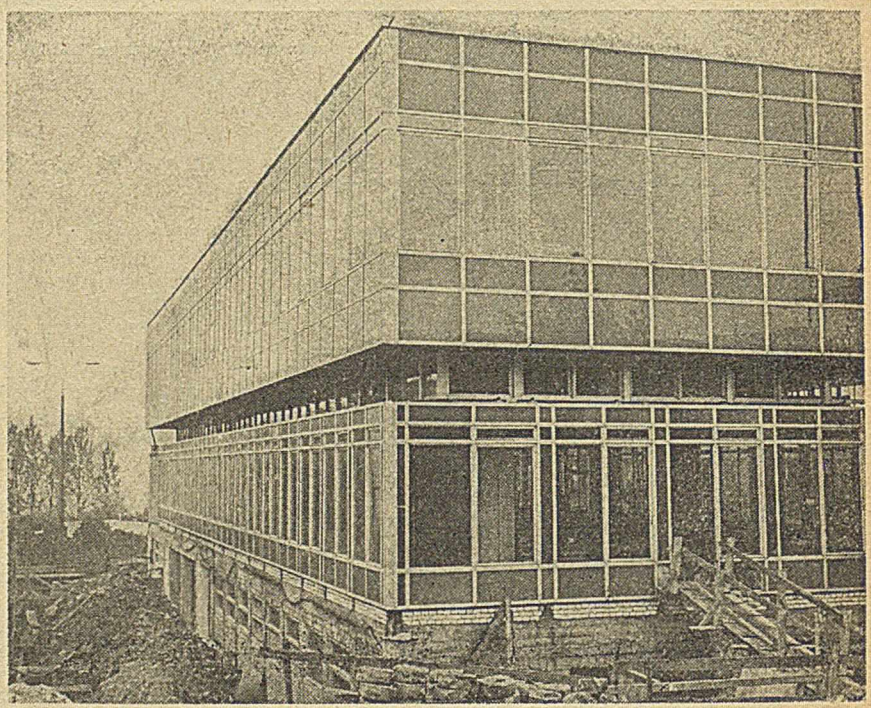
Krystyn Bernatowicz
(Foto: CAF)



Zwiększenie liczby monitorów ekranowych przyśpieszy obieg informacji.

nowanej już wcześniej budowy pomieszczenia dla maszyny i personelu ośrodka. Przeprowadzana obecnie na dużą skalę modernizacja Fabryki uniemożliwia instalację terminali, będących niezbędnym urządzeniem dla wzrostu efektywności funkcjonowania systemu. Należy się jednak spodziewać, że w pierwszych miesiącach 1975 r. oddany zostanie do użytku nowoczesny pawilon dla Ośrodka, a okres prowizorium przejdzie do historii.

A oto budynek Ośrodka w październiku 74 r. Prawdopodobnie wiosną stanie się terenem codziennej pracy informatyków Zakładu.



i zainstalowanie jeszcze jednej drukarki. Zdaje się że nastąpi to w nadchodzącym roku. Dość pilną wydaje się sprawa przesyłania danych z satelitarnych zakładów z poza Starachowic. Jak dotychczas przygotowuje je tylko Zakład w Końskich. Ale w niedalekiej przyszłości Ośrodek sceduje te zadania na pozostałe zakłady — początkowo będzie to zwrotna spedycja danych i wyników transportem konwencjonalnym a następnie po uzyskaniu i zainstalowaniu końcówek — poprzez teletransmisję.

Poważną komplikacją tak dla zamknięcia realizacji systemu jak i dla pracy zespołu Ośrodka Przetwarzania Informacji są problemy związane z inwestycjami. Przede wszystkim Ośrodek tęsknie oczekuje ukończenia budowy pomieszczenia. Tymczasowo Ośrodek jest sublokatorem na terenie zakładowej stołówki. Ograniczenia inwestycyjne wprowadzone w 1970 roku nie pozwoliły rozpocząć zapla-

Komputer w badaniach techniczno-eksploatacyjnych transportu samochodowego

W związku z rosnącym w szybkim stopniu potencjałem i zadaniami transportu samochodowego niezbędne jest prowadzenie ścisłej i skoordynowanej analizy wyników badań technicznych i eksploatacyjnych w tym transporcie. Jest to możliwe tylko przy pomocy wszechstronnego wykorzystania informatyki.

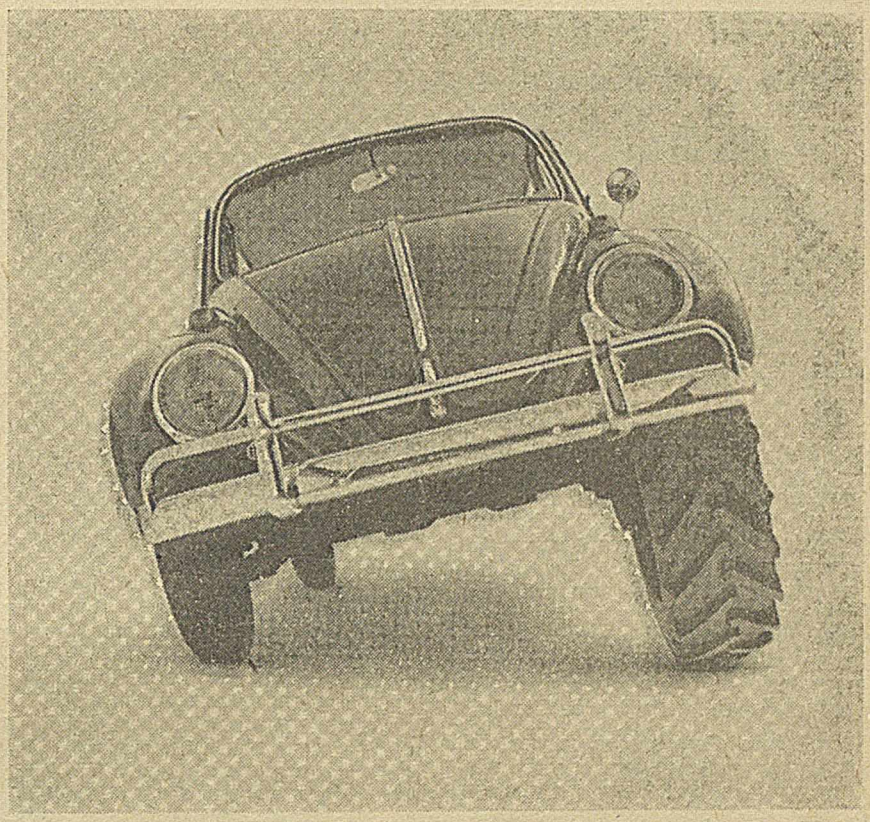
Badania w transporcie samochodowym można podzielić na techniczne i eksploatacyjne. Badania techniczne są przeprowadzane przez specjalne zespoły badawcze odpowiednich instytutów naukowych na wybranych egzemplarzach pojazdów. Badania takie przeprowadza się dwojako: w laboratoriach-hamowniach i na specjalnie dobranych odcinkach dróg kołowych. Badania eksploatacyjne występują już w ramach procesu przewozowego realizowanego przez poszczególne przedsiębiorstwa i zakłady transportu samochodowego.

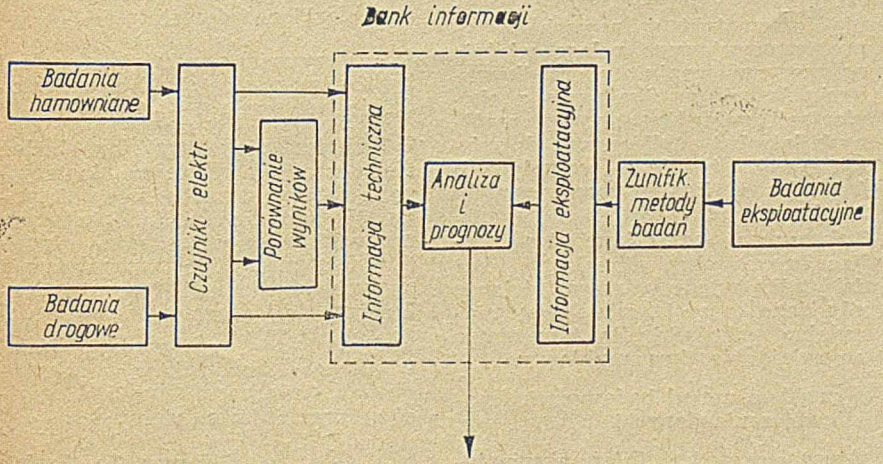
Badania eksploatacyjne są prowadzone ex post (na podstawie kart drogowych itp.) bądź też metodami bezpośredniej ingerencji w proces przewozowy (metoda kordonowa).

ZADANIA INFORMATYKI W BADANIACH TECHNICZNYCH TRANSPORTU

Obecna technika pomiarowa daje badaczom do dyspozycji szereg nowoczesnych metod pozwalających na miarodajną ocenę zjawisk występujących w silnikach i pojazdach samochodowych podczas ich pracy. Należą do nich m. in. metody opierające się na zastosowaniach elektroniki.

Pierwszy problem jaki pojawia się przy zastosowaniu maszyn elektronicznych do obliczeń i analizy zjawisk termodynamicznych, cieplnych, wytrzymałościowych i akustycznych w pojazdach samochodowych to problem przystosowania wyjść czujników elektronicznych lub mechanicznych do współpracy z komputerami. Duży procent błędów powstaje właśnie na styku wyjścia czujnika pomiarowego — komputer, bowiem częstokroć wyniki pomiarów są zapisywane w postaci wykresów na taśmach (np. przy korzystaniu z oscylografów pętlicowych) bądź kartach papieru (np. rejestratory X-Y). Niejednokrotnie czujniki, a nawet całe urządzenia pomiarowe (są to często prototypy) są projektowane bez uwzględnienia możliwości ich współpracy w przyszłości z komputerami. Często zdarza się, że nawet w przypadku teoretycznej możliwości bezpośredniego sprzężania aparatury pomiarowej z maszyną cyfrową (z technicznego punktu widzenia), w związku z zajęciem maszyny przez inny zespół badawczy, możliwość taka jest niewykorzystywana.





Rys. 1. Informatyka w badanich transportu samochodowego

W takim przypadku pojawia się problem odczytu danych pomiarowych z wykresu bądź z tabeli. Taka procedura jest uciążliwa i nie zawsze dokładna, a poza tym pracochłonna. Stąd też zainteresowanie szeregu instytutów badawczych problematyką automatycznego odczytu danych z wykresów i tabel. Prace w tym zakresie (odczyt z taśmy wskazań elektronicznego czujnika indukcyjnego przez komputer przy zastosowaniu czytnika własnej konstrukcji) były prowadzone m. in. w Laboratorium Elektroniki ZTiES Politechniki Szczecińskiej podczas badań zawirowania ładunku powietrza w

Tabela

Specyfikacja badań w transporcie samochodowym z punktu widzenia informatyki

Pozycja	Badania techniczne	Badania eksploatacyjne
Charakter badanych przebiegów	Szybkozmienne	Powolne
Wielkości mierzone	Ciśnienia temperatury, prędkości, zużycia, naprężenia	Praca przewozowa, przebieg, częstość napraw
Metoda pomiarowa	Czujniki elektryczne, mechaniczne, pneumatyczne	Metoda ankietowa, kordonowa itp.
Wyniki badań są uzyskiwane w postaci	Wykresów, tabulogramów	Ankiet, tabel
Mnogość wyników zależy głównie od:	Dokładności badań	Ilości taboru i zakresu badań
Dopływ informacji do maszyny cyfrowej	Zależy od planu badań z reguły przerywany nieregularny	Ciągły lub przerywany regularny

przestrzeni nadtłokowej silników wysokoprężnych. O ile odczytywanie automatyczne danych drukowanych i pisanych (liter i cyfr) jest stosunkowo dobrze rozpracowane (systemy OCR, OMR, MICR), to odczyt linii krzywych (np. wykresów temperatury elementów silnika) nastręcza jeszcze sporo trudności.

Wykorzystanie automatycznych metod odczytu wyników pomiarów pozwala na znaczne skrócenie, czasu ich opracowania i analizy. Ważnym problemem wynikającym z włączenia komputerów do badań techniczno-eksploatacyjnych w transporcie samochodowym jest problem efektywności wykorzystania czasu pracy maszyny. W związku z tym, że obecnie część — a w przyszłości cała aparatura pomiarowa będzie w stanie bezpośrednio współpracować z komputerami, istotnym jest optymalne zaplanowanie badań. Szczególnie jaskrawo przedstawia się ten problem w badaniach pojazdów samochodowych. Jak wiadomo — wybrane silniki poddaje się na hamowniach próbom, trwałym — trwającym po kilkadziesiąt godzin.

Współpraca komputera przez tak długi okres czasu w sposób nieprzerwany z aparaturą pomiarową byłaby niesłychanie kosztowna i mijałaby się z celem. Dlatego też istotnym jest takie planowanie prób silnikowych i samochodowych by były one jednocześnie mało czasochłonne i miarodajne. Osiąga się to m. in. przez zastosowanie ustawiacza programu (program *setter*). W badaniach tego typu czas współpracy z maszynami musi być różnicowany następująco:

- czas współpracy jest wielokrotnością jednego cyklu badanego (np. w przypadku badania przebiegu wydzielania się ciepła podczas jednego cyklu pracy silnika jest to czas rzędu 10^{-3} sekundy)

- czas współpracy jest ułamkiem jednego cyklu badań (np. w przypadku badań długotrwałych istotne jest wyodrębnienie punktów węzłowych charakterystycznych dla danego procesu i skoncentrowanie na nich większości czasu współpracy z komputerem lub rejestracji przebiegu).

ZADANIA INFORMATYKI W BADANIACH EKSPLOATACYJNYCH TRANSPORTU SAMOCHODOWEGO

O ile badania techniczne (na etapie konstrukcyjno-rozwojowym) są związane z pomiarami zmiennych będących funkcjami olbrzymiej liczby różnorodnych czynników (ciśnienia, temperatury, wilgotności, gęstości, jednorodności, przewodności, prędkości itp.) i wymagają bardzo precyzyjnych czujników elektronicznych, to badania eksploatacyjne mają nieco inny charakter. Czynników wpływających na wyniki jest tu znacznie mniej (są to z jednej strony czynniki wynikające z konstrukcji pojazdów, z drugiej strony — czynniki związane ze strukturą ładunków itp.), lecz liczba wyników pomiarów (zużycia paliwa, ogumienia, czasu pracy, liczby przewiezionych ładunków itp.) jest wielokrotnością liczby eksploatowanych pojazdów (a więc rzędu milionów) w skali kraju. Oczywiście metody statystyczne pozwalają na określanie niezbędnej populacji minimalnej, są to jednak tylko metody przybliżone, co można wywnioskować z dotychczasowych, częstokroć bardzo odbiegających od rzeczywistości prognoz transportowych. Zadanie informatyki w ramach systemu transportowego polega właśnie na magazynowaniu uporządkowanych danych wynikających z eksploatacji pojazdów w poszczególnych latach i analizowaniu dotychczasowych trendów rozwojowych (np. drogą konkretyzacji modelu Markowa).

Dokonywana przez komputer analiza danych eksploatacyjnych będzie tylko wtedy pełnowartościowa, gdy dane te zostaną ujęte w miarodajne wskaźniki. Na tym etapie pojawiają się dwa problemy:

- problem metod badawczych (sposób zbierania danych)

- problem wskaźników eksploatacyjnych.

Tak metody badań eksploatacyjnych jak i wskaźniki eksploatacyjne winny być w dużym stopniu ujednolicone i odpowiadać założeniom stymulującym rozwój zintegrowanego systemu transportowego.

Woluntaryzm w dotychczas stosowanych metodach badawczych w eksploatacji pojazdów doprowadzał do powstawania częstokroć nawet 50 procentowych błędów oszacowań. Wprowadzenie do komputerów mało odpowiedzialnych (pod względem merytorycznym) danych doprowadza siłą rzeczy do błędnych analiz i prognoz a winą za ten stan rzeczy obarcza się często informatykę.

Odnośnie wskaźników eksploatacyjnych (np. wskaźników pracy przewozowej) należy zauważyć, że często nie odzwierciedlają one prawidłowo dynamiki i istoty mechanizmu zjawisk transportowych.

Do zadań informatyki w zakresie transportu winno należeć w pierwszym rzędzie: gromadzenie danych o pracy taboru, analiza rozwiązań konstrukcyjnych z punktu widzenia eksploatacji, korelacja tych rozwiązań z danymi eksploatacyjnymi, określanie tendencji rozwojowych w zakresie potrzeb transportowych i stymulowanie (w oparciu o rachunek kosztów w okresach przyszłych) rozwoju nowych konstrukcji samochodowych. Aczkolwiek zadania stawiane przed informatyką w zakresie potrzeb transportu są olbrzymie, to przy odpowiedniej organizacji pracy można im sprostać.

Punkty węzłowe problemu, które należy w pierwszej kolejności rozwiązać to:

- unifikacja metod badawczych i rewizja przydatności aktualnie stosowanych wskaźników eksploatacyjnych

- zapewnienie rytmicznego dopływu informacji z zakresu transportu do centralnego banku danych

- określenie wzajemnych powiązań funkcjonalnych poszczególnych gałęzi transportu oraz transportu jako całości i pozostałymi gałęziami gospodarki narodowej

- opracowanie programów analizy i prognozowania zjawisk w dziedzinie transportu

- Skorelowanie wyników badań technicznych i badań eksploatacyjnych taboru samochodowego i kalkulacja opłacalności (na podstawie porównania kosztów minimalnych i maksymalnych) wprowadzenia nowych konstrukcji taboru i rozbudowy sieci drogowej.

Zadania te wymagają ścisłej współpracy konstruktorów, ekonomistów (transportowców i drogowców) oraz informatyków.

Benon Zb. Szałek
Politechnika Szczecińska

Stacja transmisji danych ICL 7503

W początkowym stadium rozwoju transmisji danych i systemów teleinformatycznych podstawową funkcją zdalnych urządzeń końcowych (terminali) był transfer danych do — lub z centrum wyposażonego w maszynę cyfrową o dużej mocy obliczeniowej. Pojawiające się wymagania na wielofunkcyjność oraz umożliwienie sterowania wykonaniem przesłanych programów doprowadziły do łączenia różnych typów terminali w duże grupy oraz wspólnego ich sterowania i buforowania.

Następny krok uczyniono po przeanalizowaniu przetwarzań wykonywanych przez komputer główny, a pochodzących z urządzeń końcowych. Okazało się mianowicie, że:

● Część przetwarzań — nieskomplikowane problemy obciążające centrum systemu typu: tabulacja, formowanie danych itp. — mogłaby być wykonana przez minikomputer o stosunkowo prostej organizacji.

● Straty czasowe związane z powtórzeniami transmisji, korekcją danych wejściowych itp. wynikają w głównej mierze z przesyłania danych niesprawdzonych — z błędami (najczęściej ortograficznymi — możliwymi do poprawienia).

Aby usprawnić pracę całego systemu przyjęto koncepcje wielofunkcyjnej programowalnej stacji transmisji danych umożliwiającej:

— transmisję danych „do" i „z" komputera macierzystego
— zdalne sterowanie wykonaniem przesłanych programów *(remote job control)*
— lokalne programowanie i wykonywanie programów
— wstępną obróbkę danych, obejmującą sprawdzenie, korekcję i ewentualny zapis do pojemnej pamięci pośredniczącej.

Reprezentantem takiego podejścia jest stacja transmisji danych 7503 produkowana przez firmę ICL.

Charakterystyka sprzętu

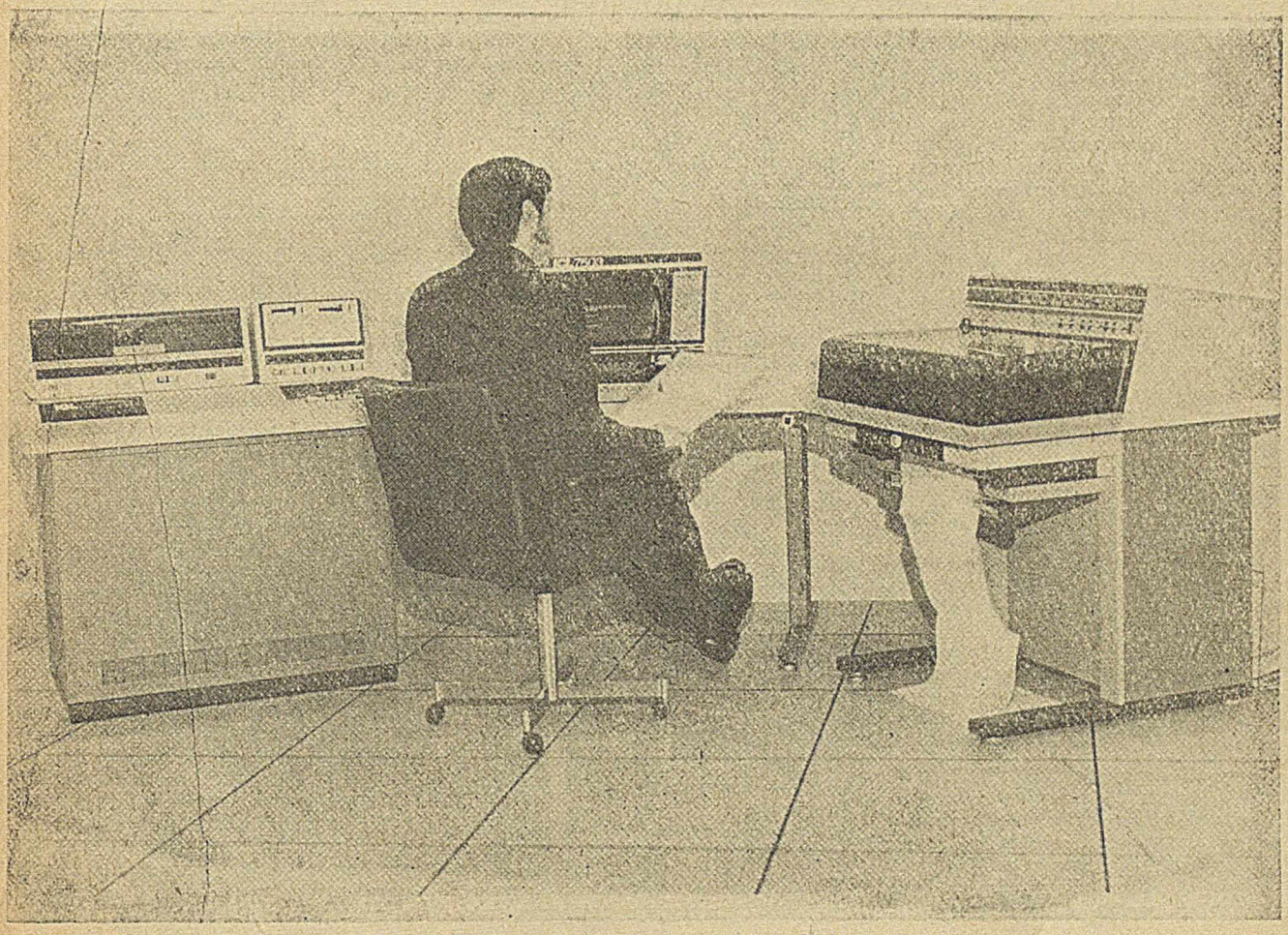
Ogólna struktura stacji transmisji danych ICL 7503 pokazana jest na rys. 1.

Jednostką centralną (JC) wykonującą wszystkie podstawowe operacje przetwarzania oraz sterującą pracą terminala jest jednoprogramowy procesor.

Zawiera on pamięć operacyjną (S) typu MOSFET o cyklu 1 μs i pojemności od 16 do 64 bajtów zorganizowaną w słowa dwubajtowe.

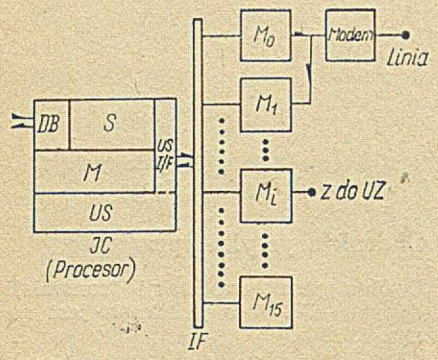
Dołączony do pamięci kanał dostępu bezpośredniego (DB) umożliwia współpracę systemu z pamięciami dyskowymi lub taśmowymi. Szesnastobitowy arytmometr (M) zbudowany z układów scalonych sumatorów typu *„lock ahead"* posiada ich standardowy repertuar

Foto: Stacja transmisji danych ICL 7503, której dostawa jest oczekiwana w tym roku dla Ośrodka GUS

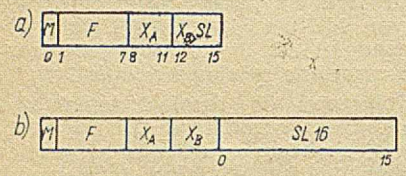


funkcji logicznych i arytmetycznych.

Średni czas wykonywania instrukcji wynosi 2 μs.



Rys. 1. Struktura urządzenia końcowego ICL 7503



Rys. 2. Postać instrukcji

Informacje pomiędzy jednostką centralną a urządzeniami zewnętrznymi przesyłane są za pośrednictwem wspólnego, dwukierunkowego interfejsu (IF), do którego dołączone są moduły (M_0 — M_{15}) sterujące pracą urządzeń zewnętrznych i pośredniczące w przekazywaniu informacji. Maksymalnie może być 16 takich modułów przy czym moduły: zerowy, pierwszy i piętnasty mają wyposażenie obowiązkowe złożone z nadajnika i odbiornika segmentu teletransmisyjnego oraz modułu procesora. Podstawowymi funkcjami modułów jest: przyjmowanie rozkazu transferu, zdekodowanie i porównanie jego ewentualnego adresu z własnym (występują również bezadresowe rozkazy transferu) oraz jego wykonanie (przesyłając dane lub statusy).

Przyjęto zasadę, że moduł odpowiada na każdy rozkaz współpracy, a jego odpowiedź może być dwojaka: w postaci danych, jeśli sterowanym przez niego jest urządzenie wejściowe lub dwunastobitowy status. Tak zaprojektowany system zapewnia znaczną elastyczność, ponieważ dołączanie urządzenia zewnętrznego odbywa się przez dołączanie właściwego dla niego modułu.

Oprogramowanie

Minikomputer ICL 7503 podczas pracy w trybie lokalnym może wykonywać programy napisane w języku CAMP. Jest to język typu assembler bardzo silnie ukierunkowany maszynowo o adresach i rozkazach zapisanych szesnastkowo i operujący na następujących elementach: zawartościach akumulatorów i komórek pamięci, stałych alfanumerycznych, bajtach, przerzutnikach warunków, wyznacznikach programowych i przerzutnikach modułów. Ogólna postać instrukcji jest następująca:

X_A, X_B — czterobitowy adres jednego z ośmiu akumulatorów adresowanych 0—7
SL — czterobitowa stała alfanumeryczna
SL1 — sześciobitowa stała alfanumeryczna
S — adres pierwszego bajtu słowa w pamięci

W jednostce centralnej zawarty jest 256 pozycyjny rejestr wskaźników programowych, które mogą być gaszone lub zapalane przez program a służą do komunikacji między rozdzielnymi częściami (poziomami) programu.
Nowym rozwiązaniem są przerzutniki modułów, dzięki którym można za pomocą rozkazów sterować pracą urządzeń zewnętrznych. Przykładowo możliwe jest programowe przydzielanie i zwalnianie urządzeń zewnętrznych, ustawianie statusów, generowanie przerwań itp.
Siedmiobitowa część rozkazowa instrukcji pozwala zakodować 128 różnych operacji. W języku CAMP część z nich jest niewykorzystana. Repertuar rozkazów jest zorientowany na sterowanie pracą urządzeń zewnętrznych i składa się z następujących grup:

Tabela I

Program: wydruk znaku na dalekopisie

Adres instrukcja		Opis
POZIOM 0		
0	630E	wybranie modułu o adresie 14(Eg14) — jest to moduł drukarki dalekopisu
2	7900	zapalenie przerzutnika 9 w grupie 0 — oznacza to:
4	0220	system gotowy. Drugie słowo rozpisane binarnie ma postać: 11 10 9 8 7 6 5 4 3 2 1 0 / 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 / przerzutnik (pozycyjnie) akcja grupa
6	6201	do akumulatora 0 wpis z grupę 1 odpowiedzi — status jeśli bit 9
8	EC90	zawartości akumulatora 0 jest zapalony, to skocz do adresu 6
A	0006	(bit. 9 = 0 statusu oznacza, że dalekopis pracuje)
C	7900	zapal przerzutnik 11 w grupie 2 (uruchom silnik dalekopisu).
E	0882	
10	7600	
12	001C	ustaw adres przełączenia poziomów = 1C
14	6800	przełącz poziom (adres przełączenia ustawiony poprzednio)
16	0000	
18	3800	
1A	0028	skok bezwarunkowy do adresu 28
POZIOM 1		
1C	0380	zezwolenie na przerwania programowe
1E	7900	zapal przerzutnik 9 w grupie 2 (zezwól na wzbudzenie) linii interfejsu („przerwanie dla pisania")
20	0222	
22	0200	zgaś wyznacznik 0
24	3B00	jeśli wyznacznik 0 zgaszony, to skocz do adresu 24 — oczekiwanie na przerwanie w pętli
26	0024	
POZIOM 0		
28	6201	jak rozkaz w komórce o adresie 6
2A	BCA0	jeśli bit 10 (A-10) akumulatora 0 równy jest 1 skocz do 28 (bit 10 statusu znaczy: dalekopis gotowy do pisania)
2C	0028	
2E	7800	wyślij do dalekopisu znak 41 (41-A)
30	0041	
32	3700	STOP

Pojawiające się w poziomie 1 przerwanie spowoduje przełączenie programu do poziomu 0, w którym zostanie ono obsłużone. Przełączenie do rozkazu następnego po 6800.

Objaśnienia

1. Poziom 0 — część programu, w której program startuje i wykonuje wstępne czynności względem całego systemu oraz obsługę przerwań
2. Poziom 1 — część programu, w której wykonywane są czynności szczegółowe odnoszące się do poszczególnych urządzeń zewnętrznych oraz uaktywniany jest system przerwań
3. Akumulatory, rejestry warunków są oddzielne dla obu poziomów i wzajemnie niedostępne. Wspólne, komunikujące oba poziomy są wyznaczniki programowe i komórki pamięci.

— instrukcje logiczne: I, LUB, NIE, SUMA MODULO DWA, porównania, przesunięcia, konwersje liczb
— instrukcje arytmetyczne: sumowanie, mnożenie, przesunięcia
— instrukcje systemowe i sterujące: skoki, transfery danych i statusów, sterowanie wyznacznikami programowymi i przerzutnikami modułów.
Możliwa jest modyfikacja ($M = 1$) rozkazów zawartością akumulatora wskazanego częścią X_B.
Pisanie programów w tym języku, jak w każdym języku niższego rzędu, jest pracochłonne i dosyć skomplikowane. W tabeli I podano przykładowy program prostego przesłania.

Istniejąca zewnętrzna (symboliczna) wersja tego języka wymaga użycia komputera macierzystego w celu wykonania kompilacji.

Wiele firm przyjęło podobną koncepcję stacji transmisji danych w oparciu o minikomputer (przykładem jest stacja transmisji danych firmy GENERAL ELECTRIC GEC 2050), ale rozwiązanie firmy ICL zasługuje na uwagę zarówno ze względu na większą elastyczność konfiguracji jak i możliwości współpracy z różnymi komputerami. Stacja transmisji danych ICL 7503 przystosowana zasadniczo do współpracy z komputerami ICL serii 1900 lub Systemu 4, pracującymi pod nadzorem właściwych systemów operacyjnych, ma również możliwość przeprowadzania emulacji umożliwiającej współpracę z maszynami cyfrowymi innych typów.

Andrzej Bylicki
Zbigniew Stefański

We francuskim przemyśle chemicznym

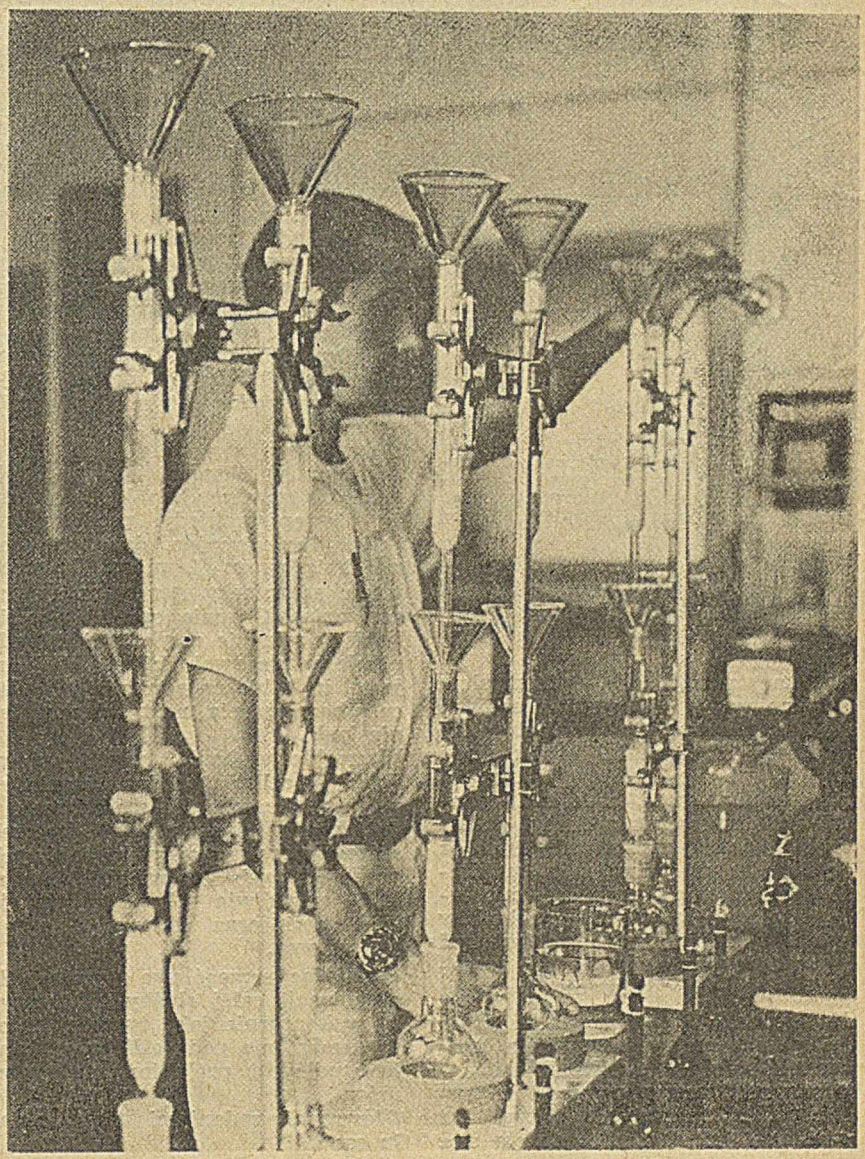
W ramach współpracy między polskim i francuskim przemysłem chemicznym prowadzona jest od 1972 r. wymiana doświadczeń w dziedzinie informatyki.

W ramach tej współpracy w czerwcu 1974 r. przebywał w Polsce dyrektor Służby Informatyki francuskiej firmy chemicznej RHONE-PROGIL, który przedstawił interesujące dane o aktualnym stanie i tendencjach rozwojowych informatyki we Francji oraz reprezentowanej przez niego firmie. Wykorzystuje ona liczny i nowoczesny sprzęt komputerowy (IBM, SIEMENS) z teletransmisją i posiada wieloletnie doświadczenia w opracowywaniu, wdrażaniu i eksploatacji systemów informatycznych.

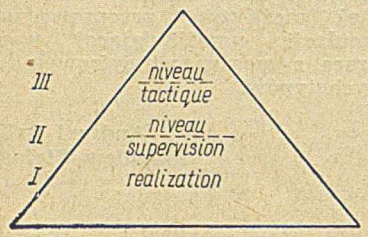
Przy opracowywaniu 3-letnich planów (kroczących) rozwoju informatyki w koncernie RHONE-PROGIL współpracują ze sobą trzy wyróżnione grupy specjalistów. Rysunek 1 przedstawia współudział tych grup i zakres ich odpowiedzialności:

Przy opracowywaniu banku danych zachowuje się zasadę, że użytkownicy z poszczególnych przedsiębiorstw mają możliwość indywidualnego dostępu do zbiorów danych. W centrum obliczeniowym firmy w Paryżu eksploatuje się głównie programy systemu dotyczące potrzeb wszystkich użytkowników. Poza tym w filialnych ośrodkach obliczeniowych poszczególne przedsiębiorstwa eksploatują własne programy.

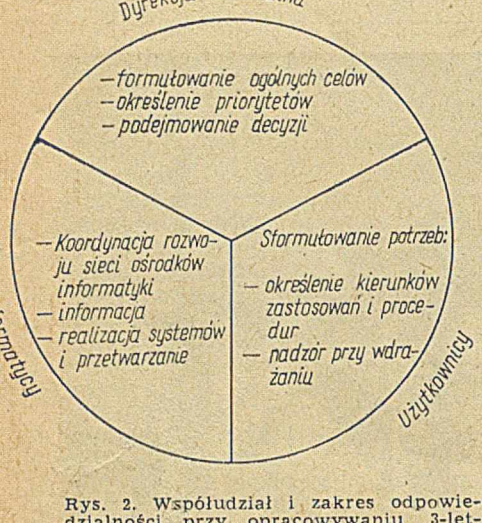
Instnieje przy tym tendencja, aby nie dopuszczać do opracowywania wielkich systemów zintegrowanych ze względu na ich znaczne zdezaktualizowanie się w momencie ich ukończenia.



Główne kierunki zastosowań informatyki to zarządzanie produkcją. Inne dziedziny takie jak np. transport, czy finanse traktuje się jako służebne w stosunku do produkcji.



Rys. 1. System informatyczny zarządzania koncernem:
I — poziom systemów operacyjnych t.j. systemów bieżącego zarządzania przedsiębiorstwem
II — poziom systemów nadzorczych działalności podstawowej wyznaczonej przez plany taktyczne i strategiczne
III — poziom systemów taktycznych i strategicznych



Rys. 2. Współudział i zakres odpowiedzialności przy opracowywaniu 3-letnich planów rozwoju informatyki we francuskim koncernie chemicznym

System zarządzania przemysłem ekonomicznym jest trójszczeblowy. Na I-szym poziomie eksploatuje się systemy zarządzania produkcją i związane z nimi systemy zarządzania pracownikami, zapasami, księgowości i utrzymania ruchu. Kadra kierownicza korzysta nie tylko z wydruków ale również z monitorów ekranowych, które są zainstalowane w gabinetach dyrektorów szczebla operacyjnego, (gdyż oni zarządzają produkcją), a nie w gabinetach dyrekcji koncernu.

Na II-im poziomie funkcjonują jednolite systemy zarządzania koncernem zasilane informacjami odpowiednio zagregowanymi z systemów zakładowych. Dlatego też w firmie jest silnie rozwinięta sieć transmisji danych, która łączy komputery ośrodków filialnych z komputerem ośrodka centralnego.

Na III-cim poziomie zarządzania eksploatuje się absolutnie jednolite systemy w koncernie dla potrzeb dyrekcji generalnej, takie jak np. zarządzanie kadrami, ustalenie poziomu budżetu rocznego itp.

Informatyka jest pomocnym narzędziem w zarządzaniu na szczeblu strategicznym dzięki modelom symulacyjnym umożliwiającym prawidłowe przewidywanie. Firma ma wiele programów symulujących np. wyniki działalności przedsiębiorstwa (system CONCORDE). Symulację przeprowadza się na danych hipotetycznych. Ważniejsze modele to: model pozwalający prowadzić elastyczną politykę płac, bardzo pomocny np. w negocjacjach ze związkami zawodowymi oraz model symulowania budżetu przedsiębiorstwa pozwalający badać wpływ podwyżki cen surowców na ceny wyrobów gotowych. Wykorzystywanie w/w modeli umożliwia optymalizowanie wielkości produkcji. Obliczenia realizuje się szybko przy pomocy bardzo małych programów.

Banki danych wielkich systemów są zbiorem informacji o wszystkich zjawiskach występujących w przedsiębiorstwie. W wyniku sukcesywnego wdrożenia kolejnych odcinkowych systemów w poszczególnych przedsiębiorstwach zbiera się te same informacje, które są potrzebne do zasilania banków wielkich systemów. Powstał więc problem ewentualnej niezgodności danych np. indeksów w zakresie tego samego tematu. Dlatego też buduje się odrębne specjalizowane banki dla poszczególnych dziedzin działalności, zachowując zasadę:

a) „niezaśmiecania" banków tzn. zbierania tylko takich danych, które są potrzebne wielu użytkownikom

b) szybkiego dostępu do danych.

Jest już w eksploatacji np. bank środków trwałych, do którego mają dostęp: zakłady produkcyjne, służba inwestycyjna, dyrekcja pionu księgowego i pionu finansowego, sekretariat generalny (ubezpieczenia). Obecnie opracowuje się bank w zakresie gospodarki zatrudnieniowo-płacowej, który obejmuje m. in. takie zagadnienia jak płace, ubezpieczenia, renty, doskonalenie kadr.

Wszystkie te banki tworzą system IMS. Umożliwia on korzystanie ze zbiorów danych, ich modyfikację i pobieranie danych dla potrzeb szczegółowych systemów własnych poszczególnych przedsiębiorstw. IMS umożliwia również selekcję informacji oraz zadawanie pytań do systemu.

Pomijając szczegółowe omówienie organizacji całej służby informatyki w koncernie jako typowej należy szczególnie podkreślić rolę działu studiów i projektowania systemów.

Głównymi zadaniami tego działu są:

- prowadzenie prac badawczych, projektowanie i wdrażanie nowych systemów
- aktualizowanie istniejących systemów
- zagwarantowanie łączności z analogicznymi działami studiów w filialnych ośrodkach użytkowników.

Do działu studiów należą:

- zespoły projektantów systemów i programistów d/s systemów takich jak: zarządzanie kadrami, sprzedaż, produkcja i jej wyniki, zaopatrzenie i transport, księgowość — finanse, analiza działalności przedsiębiorstwa. Każdy zespół projektuje system od początku, sukcesywnie wdraża kolejno przygotowane podsystemy i odpowiada za jego rozwój. Tego rodzaju opieka nad losami zaprojektowanego przez siebie systemu daje doskonałe rezultaty, gdyż zwiększa odpowiedzialność projektantów i zapewnia ich stałą łączność z użytkownikami.
- komórka koordynacji zobowiązana do zabezpieczenia powiązań pomiędzy systemami na etapie projektowania
- komórka metod i procedur administracyjnych

Tabela I

Struktura kosztów informatyki w koncernie chemicznym w 1973 r.

Kierunki zastosowania systemów informatycznych	Udział w kosztach %
— zarządzanie	86,0
— obliczenia naukowo-inżynierskie	8,0
— analiza chemiczna (instrumentalna)	5,0
— wyszukiwanie dokumentacji	0,8
— inne	0,2
Ogółem koszty informatyki	100,0

Tabela II

Struktura kosztów informatyki w zakresie zarządzania

Kierunki zastosowania systemów	Udział w kosztach %	Uwagi
— działalność handlowa	39	
— finanse i księgowość	15	
— zarządzanie kadrami	6	z tendencją wzrostową
— zarządzanie produkcją	5	
— zarządzanie transportem	5	
— zarządzanie przedsiębiorstwami filialnymi	9	
— inne	21	
Ogółem APD	100	

● komórka planowania i rozwoju szczególnie zobowiązana do:
a) opracowania planu informatyki przy współpracy z Komisją do Planu Informatyki
b) rozwoju systemów informatycznych przy współpracy z innymi przedsiębiorstwami zgrupowanymi w koncernie RHONE-POULENC i filialnymi z RHONE-PROGIL.
Pozostałe działy Służby Informatyki są następujące:

● dział systemów i metod zajmujący się sprawami sprzętu i oprogramowania teletransmisji oraz ogólną metodologią projektowania i programowania

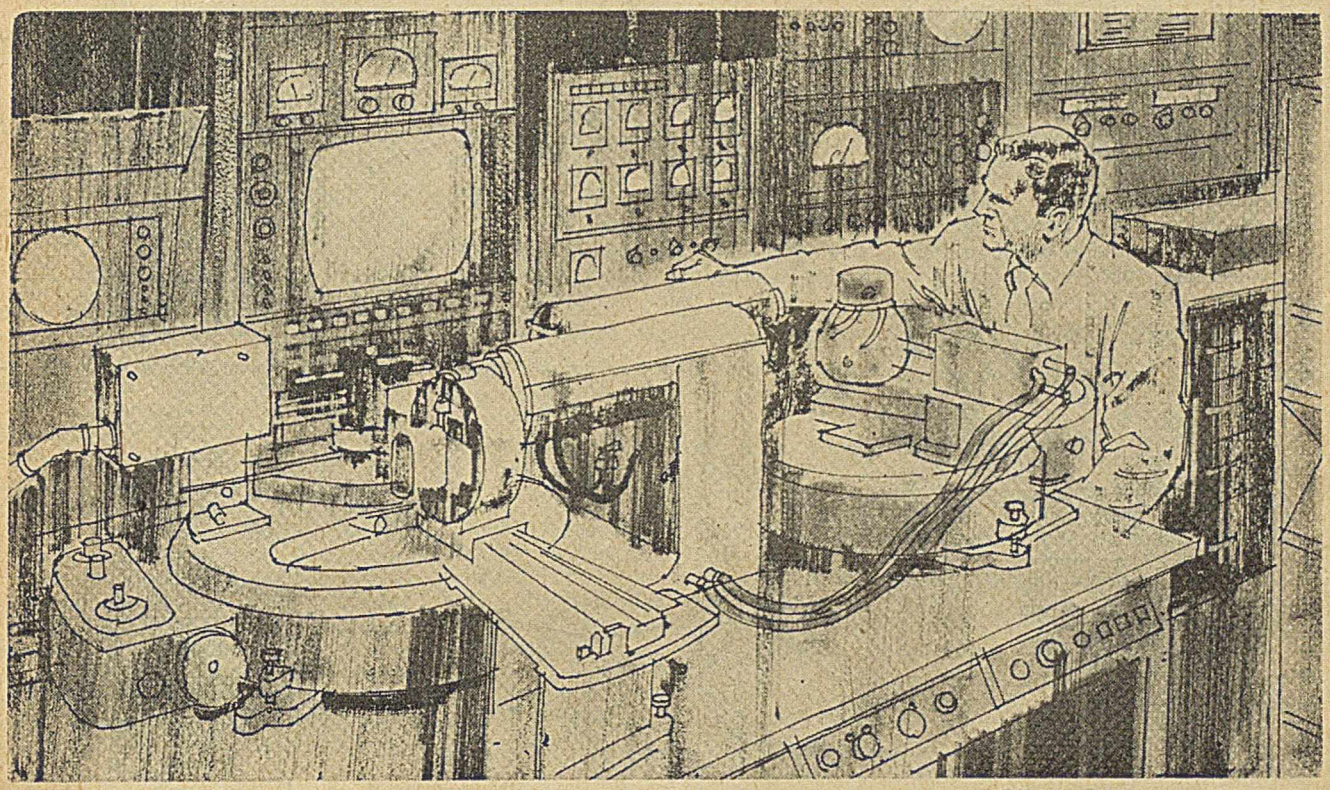
● dział „decentralizacji informatyki", którego zadaniem jest pomoc przedsiębiorstwom filialnym na miejscu przy zastosowywaniu środków technicznych informatyki. Jest to ważna funkcja, która jest zrealizowana pod kątem zabezpieczenia polityki decentralizacji określanej przez kierownictwo Służby Informatyki

● dział eksploatacji obejmujący pełny cykl przetwarzania począwszy od kontroli dokumentów wejścia

● sekcja administracyjna, której rolą jest rozdział zadań niezbędnych do wewnętrznego funkcjonowania Służby Informatyki, a m. in. zarządzanie kadrami i określanie mocy przerobowej w zakresie projektowania.
Szybko rozwijające się systemy informatyczne w zakresie zarządzania spowodowały zmiany struktur organizacyjnych w przemyśle. Służby organizatorskie połączyły się ze służbami informatycznymi jako, że prace organizatorskie (np. schemat obiegu dokumentów) stanowią integralną część projektowania systemów APD. Służby organizatorskie z punktu widzenia szyldu straciły swoją rangę, gdyż zostały niejako wchłonięte przez informatykę.

Pewne funkcje służb organizatorskich pozostały autonomiczne, ale ogólnie oceniając służby te nie mają dziś takiego znaczenia w przedsiębiorstwie jak dawniej, w przeciwieństwie do ogromnego wzrostu roli i znaczenia służb informatycznych.

Maria Trzaskowska
Ministerstwo Przemysłu Chemicznego



Zastosowania w medycynie

Instytut Badawczy i Rehabilitacyjny w Huston, Teksas w ciekawy sposób użył maszyny IBM do bardzo dokładnych, trójwymiarowych pomiarów ciała ludzkiego. Na podstawie fotografii urządzenie rozpoznające wyszukuje wspólne punkty inaczej rozmieszczone na każdym ze zdjęć i tworzy trójwymiarowy obraz. Obraz ten składa się z ponad 40 tysięcy punktów i jest zapamiętywany przez maszynę. Na każde żądanie może być wówczas poprzez urządzenia graficzne odtworzony dowolny fragment ciała w różnych rzutach, drukowane tablice zestawiające określone wymiary itp. Dane te przydają się zarówno do bieżących diagnoz, jak i do badań statystycznych w szerokim zakresie; począwszy od deformacji noworodków, po spadek wagi u kosmonautów.

Metody reprezentacji i opisu struktur danych

Jednymi z istniejących i być może najczęściej obecnie dyskutowanych problemów z zakresu systemów zarządzania bazą danych są problemy opisu logicznych struktur danych i ich reprezentacji fizycznej. Zanim przejdziemy do omówienia tych problemów musimy poświęcić nieco miejsca samemu terminowi „logiczna struktura danych" i temu, co się za nim kryje. Logiczna struktura danych *(logical data structure)* zawiera poza opisem poszczególnych danych wyrażenie logicznych współzależności między danymi. Współzależności te są także nośnikiem informacji na równi z samymi danymi. Wraz z pojawieniem się problemu opisu bazy danych powstał też problem stworzenia aparatu opisu tych współzależności.

Wyróżniamy następujące elementy logicznej struktury danych [1]:

- dana — pojęcie elementarne (identyfikacja lub miara określonej cechy obiektu)
- grupa danych (zapis — sekwencja danych i ewentualnie innych grup danych mogących powtarzać się w ramach zapisu)
- relacja grup danych — logiczne współzależności między grupami danych
- zapis — zbiór grup danych (zapisów) i ich wzajemnych relacji, w którym jeden i tylko jeden podzapis zwany kluczowym, nie jest zawarty ani podporządkowany w żadnym (żadnemu) innym (innemu) podzapisie (zapisowi)
- zbiór — zbiór zapisów
- baza danych — zbiór zbiorów mających wspólny opis bazy danych.

Każdy z tych elementów musi być zdefiniowany w opisie logicznej struktury danych. Jak widać z samego wyboru elementów struktury danych, grupy danych są tymi elementami bazy danych, między którymi wzajemne współzależności istotne są z punktu widzenia opisu bazy danych.

Zauważmy, że grupa danych może, w skrajnych przypadkach, zawierać jedną daną lub też stanowić zapis. Tak więc rozpatrywanie wzajemnych relacji tylko między podzapisami nie zawęża ogólności problemu.

W praktyce spotykamy się najczęściej z następującymi dwoma strukturami danych: [1]

- struktura drzewiasta (rys. 1a i 1b)
- struktura sieciowa (rys. 2a i 2b).

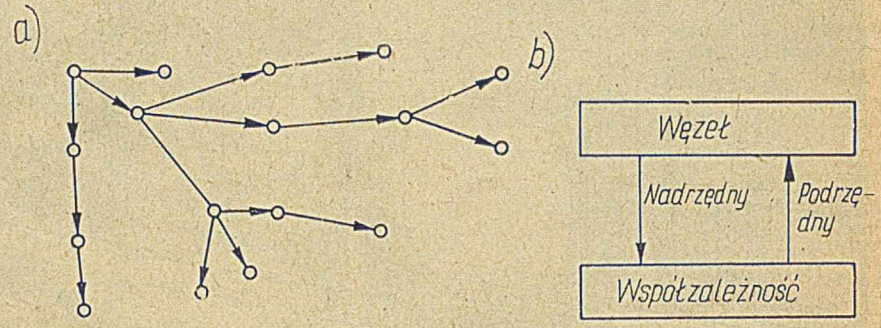
Opis bazy danych jest mechanizmem pośredniczącym między bazą danych i jej oprogarmowaniem.

Języki opisu bazy danych powinny uwzględniać następujące podstawowe wymagania:

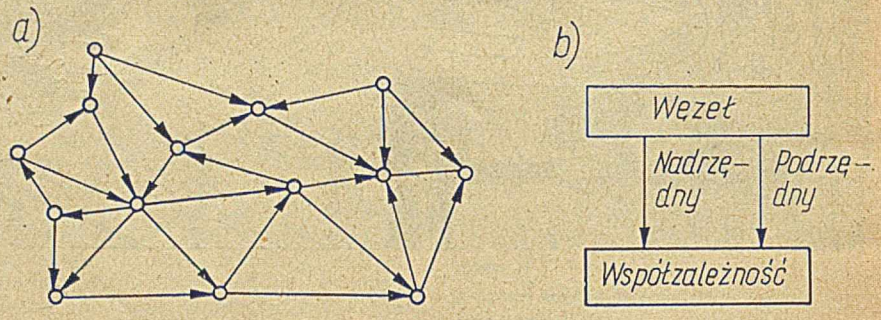
— możliwość zastosowania dostatecznie prostych języków manipulacji danymi

— możliwość zdefiniowania dostatecznej mocnej ochrony danych

— możliwość zaprojektowania fizycznej struktury danych o dostatecznie dobrych parametrach eksploatacyjnych (czas wyszukiwania, wielkość pamięci).



Rys. 1 a) Graf struktury danych typu drzewo, b) Diagram struktury danych*) typu drzewo



Rys. 2 a) Graf struktury danych typu sieć, b) Diagram struktury danych*) typu sieć

— niezależność oprogramowania od zmian w logicznej strukturze danych

— niezależność struktury logicznej od struktury fizycznej danych

— elastyczność opisywanych struktur

— prostota opisu struktur danych

— uniwersalność, tj. możliwość opisu dostatecznie złożonej struktury danych

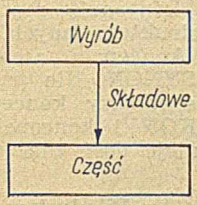
Te i wiele innych nie wymienionych tu wymagań stawianych przed językami opisu struktur danych są często wzajemnie sprzeczne i długo jeszcze chyba trwać będą dyskusje między zwolennikami różnych w tej materii koncepcji.

W obecnej chwili najbardziej godne uwagi są dwie koncepcje oparte na krańcowo różnych założeniach.

Pierwsza — reprezentowana jest przez DBTG [2] Komitetu CODA-

[1] Diagramy struktur danych zaproponowane przez C. A. Bachmana (*data structure diagrams*) są wygodnym narzędziem graficznego przedstawiania struktur danych [2].

SYL, druga — przez E. F. Codd'a z IMB.
Omówimy pokrótce podstawowe założenia tych dwu koncepcji.



Koncepcja DBTG Komitetu CODASYL

W swych kolejnych raportach z lat 1969, 1971 i 1973 Komitet CODASYL opublikował rezultaty prac swej specjalnej grupy roboczej *(Data Base Task Group)*. Raporty te zawierają specyfikacje języków:

— DDL *(Data Description Language)* — do opisu logicznych struktur danych

— DML *(Data Manipulation Language)* — rozszerzenie języka COBOL o instrukcje dostępu do bazy danych.

Założenia języków DDL/DML wynikają z przyjętej przez CODASYL koncepcji systemu zarządzania Bazą Danych. Koncepcja ta jest próbą stworzenia pomostu między dotychczasową filozofią przetwarzania danych, a filozofią nową, wymuszoną przez pojawienie się baz danych. Źródła tej koncepcji można już znaleźć w języku COBOL (zdanie COPY).

Cechą charakterystyczną tej koncepcji jest rozbudowany aparat opisu struktur danych (DDL). W języku DDL opisać można dowolnie złożone struktury danych, włącznie do struktur sieciowych zawierających cykle. Kolejne wersje języka DDL w coraz większym stopniu są niezależne od fizycznej struktury bazy danych i dokonują coraz wyraźniejszego podziału funkcji między aparatem opisu bazy danych (DDL) i aparatem manipulacji danymi (DML).

Podstawowymi, nieprzezwyciężonymi dotychczas mankamentami omawianej koncepcji są trudności, jakie pojawiają się przy próbach realizacji konwersacyjnego dostępu do bazy danych, oraz trudności napotykane przy realizacji aparatu umożliwiającego wprowadzanie zmian w opisie logicznej struktury danych już w trakcie istnienia bazy danych.

Koncepcja E. F. Codd'a

W przeciwieństwie do koncepcji CODASYL'u — koncepcja reprezentowana przez E. F. Codd'a jest w kraju mniej znana, dlatego też poświęcimy jej nieco więcej miejsca.

Nie ulega wątpliwości, że jedną z najbardziej naturalnych dla użytkownika i rozpowszechnionych form reprezentacji danych jest forma tabelaryczna. E. F. Codd uważa, że nie godzi się zmuszać użytkownika do traktowania swych danych w całej ich złożoności powiązań sieciowych, skoro ogromna większość jego problemów wcale tego nie wymaga [3]. Istnieją zastosowania takie, jak: problemy transportowe, sieci energetyczne i im podobne, których sformułowanie — w przypadku istnienia aparatu opisu sieciowych struktur danych — otrzymuje się bezpośrednio.

Zastosowania „sieciowe" stanowią jednak — wg. E. F. Codda — znikomy procent problemów w zarządzaniu, czy w zastosowaniach organizacji handlowych i produkcyjnych i mogą być rozwiązywane przez oprogramowanie specjalistyczne.

Wszystkie struktury logiczne danych, poza sieciami, dają się przedstawić w postaci tablic. Termin tablica popularnie rozumie się jako macierz prostokątna, z następującymi właściwościami *(w)*:

W_1: wszystkie elementy danej kolumny są tego samego typu

W_2: każdy element tablicy jest pojedyńczą wartością danej (liczba lub ciąg znaków)

W_3: wszystkie wiersze macierzy są różne

W_4: kolejność wierszy i kolumn jest nieistotna

W_5: każdej kolumnie może być przypisany unikalny w ramach tablicy identyfikator.

Dzięki własnościom W_3 i W_4, taki rodzaj tablic jest tym, co matematycy określają terminem „relacja". Jeśli tablica ma n kolumn, mamy do czynienia z relacją n-tego stopnia.

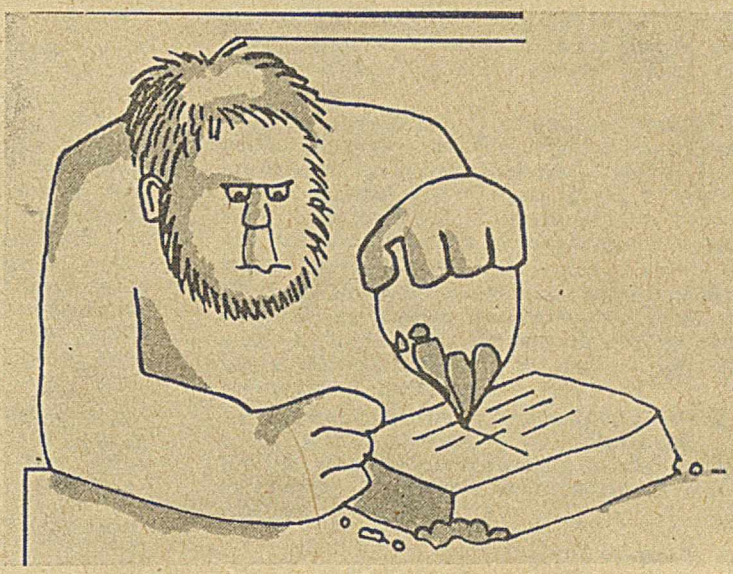
Relacja R $(D_1, \ldots D_n)$ definiuje podzbiór produktu kartezjańskiego danych $D_1, \ldots D_n$.

Rozpatrzmy prosty przykład struktury danych, którego diagram przedstawiony jest na rys. 3.
Relacja SKŁADOWE o trzech elementach, a mianowicie: numer wyrobu (NR. W), numer części (NR. C) i liczba części wychodzących w skład jednej sztuki wyrobu (IL) może mieć przykładowo następującą postać:

SKŁADOWE (NR. W,	NR. C,	IL)
1	216	2
2	817	1
2	823	6
5	216	3
5	823	2
7	300	24

Mylące jest wrażenie, że „relacyjny model bazy danych" zmusza do pewnej redundancji informacji. Pamiętać jednak należy, że mówimy wyłącznie o logicznej strukturze danych. Relacje są zbiorami abstrakcyjnymi i fizyczna organizacja bazy danych może, ale wcale nie musi, wprost odwzorowywać zbiorów — relacji.

Zdefiniowanie relacji na zbiorach bazy danych jest jednym ze sposobów opisu logicznej struktury bazy danych. Jak się okazuje, spotykane w praktyce struktury danych (oprócz struktur sieciowych) dają się w prosty sposób zdekomponować do postaci relacji, a więc tablic. Więcej, każdą ze struktur można opisać za pomocą relacji w tzw. trzeciej formie normalnej, charakteryzującej się tym, że dane wchodzące w skład relacji w sposób bezpośredni zależą od występujących w tejże relacji danych kluczowych. Jak widać w podanym wyżej przykładzie, „ilość części wchodzących w skład jednego wyrobu (IL)" zależy wyłącznie od danych kluczowych NR. W i NR. C. Opis struktury danych za pomocą relacji w trzeciej formie normalnej jest szczególnie wygodny do



[3]) Data Base Task Group — grupa robocza Komitetu CODASYL.

obsługi takich operacji na bazie danych jak: wyszukiwanie, aktualizacja, ochrona danych, ponieważ minimalizuje się w ten sposób operacje na wielu relacjach jednocześnie.

Dla relacyjnego modelu bazy danych E. F. Codd opracował język ALPHA, zawierający aparat opisu danych i manipulacji danymi [4]. Język ten pozwala w dowolnym momencie, już w trakcie eksploatacji bazy danych, skreślać stare i/lub wprowadzać nowe relacje do opisu bazy danych.

Zasadnicza różnica, w przedstawionych w ogromnym skrócie, dwu koncepcjach leży w różnej ocenie potencjalnego użytkownika.

Według koncepcji CODASYL'u użytkownik komunikować się będzie z systemem za pośrednictwem profesjonalistów, tj. administratora bazy danych i zawodowych programistów, podczas gdy koncepcja E. F. Codd'a wycelowana jest na użytkownika „aktywnego", komunikującego się z systemem bezpośrednio (np. poprzez urządzenie końcówek), choć nie wyklucza zastosowania języków bazowych *(host-languages)*.

Koncepcja CODASYL'u jest aktualnie o wiele mocniejsza, chociażby dlatego, że prace nad nią trwają od kilku lat; jest ona szeroko rozpopularyzowana a w jej tworzeniu bierze udział wiele znanych korporacji. Języki DDL/DML mają już kilka implementacji. Prawie wszystkie duże firmy komputerowe, oprócz IBM, mają już lub są w trakcie opracowywania SZBD opartych na tej koncepcji. Natomiast koncepcja E. F. Codd'a jest w obecnej chwili koncepcją tylko jego i firmy IBM.

Najmocniejszym argumentem przemawiającym za językami DDL/ /DML jest fakt, iż prawdopodobnie będą one podstawą światowego standardu w tej dziedzinie.

Koncepcja E. F. Codd'a ma jednak kilka istotnych zalet w porównaniu z koncepcją CODASYL'u a w szczególności:

— pozwala ona na zmianę opisu logicznej struktlury danych, a tym samym ewolucyjne dopasowywanie bazy danych do zmieniających się potrzeb użytkownika

— struktura logiczna w mniejszym stopniu determinuje fizyczną jej reprezentację, a co się za tym kryje, łatwiej jest zaprojektować język zapytań i taką organizację bazy danych, by zapewnić szybki proces wyszukiwania

— daje użytkownikowi możliwość „widzenia" swej bazy danych w sposób dla niego najbardziej naturalny, tj. w formie tablic.

Czytelnicy tego cyklu artykułów mieli możność zapoznać się z jeszcze jedną koncepcją reprezentacji logicznej struktur danych, a mianowicie z koncepcją zrealizowaną w SZBD SYKON. Nie trudno zauważyć, że między koncepcją systemu SYKON i koncepcją E. F. Codd'a istnieje dość znaczne podobieństwo.

Władysław Bogucki
Witold Staniszkis
ZOWAR R-POLSYSTEM

LITERATURA

[1] The CODASYL DBTG Report, 1969, 1971, 1973.

[2] CH. Bachman: Data Structure Diagrams

[3] E. F. Codd: A Relational Model of Data for Large Shared Data Banks, Comm. ACM, 1970,06.

[4] E. F. Codd: A Data Base Sublanguage Founded on the Relational Calculus, ACM Sigfidet Workshop, 1971, 11

[5] W. Bogucki, W. Staniszkis: Projektowanie Banku Informowania Kierownictwa, Problemy Informatyki, 1974.

Urządzenia WE/WY*

Szybki rozwój elektronicznej techniki obliczeniowej, stałe rozszerzanie jej zastosowań w gospodarce, nauce i technice spowodowały, że znanych jest kilkaset typów i odmian urządzeń zewnętrznych. Ich koszt osiąga w niejednym systemie liczącym 80% wartości sprzętu. Ta różnorodność i wysoki udział w kosztach, a przede wszystkim wpływ urządzeń zewnętrznych na efektywność pracy systemu, nie znajdują odbicia w naszych pozycjach wydawniczych, które poświęcone są przede wszystkim centralnym zespołom komputera.

Dlatego godna poparcia jest inicjatywa Wydawnictw Naukowo-Technicznych, które przygotowują lub już wydały książki z dziedziny urządzeń zewnętrznych. Książki te powinny być jednak aktualne i zawierać nie tylko opisy rozwiązań konstrukcyjnych, lecz uczyć organizacji procesów wprowadzania, wyprowadzania i zapamiętywania informacji. Nie mniej ważną sprawą jest ścisłość sformułowania i poprawność terminów fachowych. Niestety, wymagań tych nie spełnia recenzowana książka, której materiał pochodzi z lat sześćdziesiątych. Znajduje to wyraz nie tylko w wykazie literatury (około 40% pozycji ukazało się do roku 1965, około 50% do roku 1969), lecz przede wszystkim w opisie „historycznych" już dzisiaj konstrukcji.

Nie wydaje się celowe np. zamieszczanie opisu dziurkarki taśmy papierowej o olbrzymich gabarytach, głośnej i zawodnej, której jedyną zaletą jest wysoka szybkość pracy (300 znaków/s. — str. 56). Znaczenie tej zalety jest niezbyt istotne w warunkach współczesnej organizacji komputerów.

W wielu miejscach można mieć zastrzeżenia do jasności sformułowań. Na przykład ze zdania na str. 60 „dziurkowanie kart różni się tym od dziurkowania taśmy, że karty przechowywane w postaci pakietów (plików), umożliwiają równoległe dziurkowanie jednocześnie całej informacji..." wynika, że wynaleziono urządzenie, które potrafi jednocześnie dziurkować cały plik kart. Na str. 109 podano, że zadaniem układu komunikacji „...jest organizacja łączności między urządzeniami wejścia-wyjścia a kanałem maszyny cyfrowej". Z kolei na str. 135 znajdujemy zdanie kolidujące z definicją układu komunikacji: „kanał jest to rozwinięty, niezależny w sensie logicznym, układ komunikacji". A dwupoziomowy sygnał wizyjny otrzymał przymiotnik „czarnokolorowy" (str. 97).

Przykładami błędów terminologicznych mogą być „registry" (rejestry, str. 11), „masyw informacji" (blok informacji, str. 114), „otwór prowadnikowy" taśmy papierowej (str. 15), „informacja zafiksowana na taśmach" (str. 24), „centralny arytmometr" (procesor, str. 136).

O tym, że taka „terminologia" może prowadzić do nieporozumień, nie trzeba nikogo przekonywać.

Przejdźmy jednak do treści książki. Rozdział 1 poświęcony jest podstawowym pojęciom i klasyfikacji urządzeń zewnętrznych. Omówiono sposoby przedstawienia informacji, klasyfikację urządzeń wejściowo-wyjściowych, nośniki informacji. Niesłusznie potraktowano na równi taśmy i karty dziurkowane. Przecież 80—90% wszystkich danych wprowadza się do komputerów z kart. O taśmach, czy dyskach magnetycznych w ogóle nie wspomniano.

W rozdziale 2 autorzy książki zajęli się czytnikami/ /dziurkarkami taśm i kart. Trudno się zgodzić z zawartym tam twierdzeniem, że czytniki taśmy i kart „...umożliwiają czytanie informacji z dużą szybkością" (str. 24). A zdanie „cenną własnością tych urządzeń jest możliwość wprowadzania dużej ilości informacji" (str. 25) koliduje ze zdaniem na str. 57 „taśma dziurkowana jako nośnik informacji jest objętościowo duża i dosyć droga, dlatego też zazwyczaj nie wprowadza się na niej dużej ilości informacji".

W rozdziale 3 omówiono bardzo ogólnie urządzenia wejściowo-wyjściowe alfanumeryczne i graficzne oraz urządzenia transmisji danych. Nie powinno się jednak pisać o automatycznym czytniku rękopisu wymieniając na równi takie jego podzespoły, jak „...mechanizm przesuwu dokumentu, źródło oświetlenia, zespół odczytu i rozpoznawania znaków...". Powstaje bowiem u czytelnika, wrażenie, że problem rozpoznawania rękopisu został już rozwiązany (str. 94).

Tematem czwartego i ostatniego rozdziału jest organizacja pracy urządzeń wejściowo-wyjściowych w systemach liczących: funkcje układów komunikacji, typowe układy komunikacji, praca równoległa urządzeń wejściowo-wyjściowych, system przerwań, kanał, metody układowej kontroli pracy urządzeń wejściowo-wyjściowych.

W rozdziale tym zastrzeżenie budzi między innymi rys. 4.12, na którym dane z kanału przesyłane są do pamięci operacyjnej za pośrednictwem procesora, a nie z jego pominięciem, co stanowi zasadniczą ideę zastosowania kanałów.

Na zakończenie trzeba stwierdzić, że wiele omówionych błędów powstało z winy tłumacza.

Wbrew notce informacyjnej, z książki korzystać mogą jedynie konstruktorzy urządzeń wejściowo-wyjściowych.

Wojciech Kuczborski

*) W. I. Adaśko, J. A. Kaszczawcew, W. B. Pac: „Urządzenia wejściowo-wyjściowe maszyn cyfrowych". Tłumaczył A. Pankow, WNT, Warszawa 1974, s. 159.

Bibliografia wydawnictw polskich z dziedziny informatyki

● **Maszyna cyfrowa narzędzie człowieka współczesnego** — HAWRYLUK J. WNT, Warszawa 1974, s. 434, cena zł 44,— Informacja i jej przetwarzanie. Społeczne aspekty przetwarzania informacji. Wyposażenie techniczne maszyny cyfrowej. Programowanie. Wyposażenie programowe i praca maszyny cyfrowej. Zastaosowania maszyny cyfrowej: badania naukowe, automatyzacja zarządzania, sterowanie procesów, automatyzacja prac projektowych, obsługa masowa, ochrona zdrowia, szkolnictwo, symulacja itd. Przyszłość współpracy człowieka z maszyną cyfrową.
Książka wprowadzająca w działanie i zastosowanie maszyn cyfrowych, przeznaczona jest dla szerokiego kręgu czytelników.

● **Technika analogowa i hybrydowa** — MĘDRZYCKI J. WNT, Warszawa 1974, str. 403, cena zł 72,—
Maszyna hybrydowa. Wzmacniacze operacyjne. Zastosowania wzmacniaczy w układach regulacyjnych i pomiarowych. Generatory funkcji nieliniowych. Układy mnożące. Przetworniki analogowo-cyfrowe i cyfrowo-analogowe. Elementy hybrydowe. Programowanie. Rozwiązywanie równań różniczkowych cząstkowych. Modelowanie procesów stochastycznych. Modelowanie układów z opóźnieniem.
Książka przeznaczona jest dla inżynierów konstruktorów i użytkowników maszyn analogowych i hybrydowych oraz urządzeń pomiarowych i regulacyjnych.

● **Układy przełączające. Elementy teorii i projektowanie** — WALIGÓRSKI S. Wyd. WNT, Warszawa 1974, str. 307, cena zł 32,—
Rachunek zerojedynkowy. Elementy podstawowe i analiza układów przełączających. Wyrażenia normalne. Wyznaczanie nieskracalnych wyrażeń normalnych. Wyznaczanie optymalnych zbiorów wyrażeń normalnych. Projektowanie sieci bez ryzyka. Projektowanie sieci z przerzutnikami. Ogólne zasady projektowania układów przełączających.
Książka zapoznająca z podstawami teorii układów przełączających, z metodami ich projektowania, przeznaczona jest dla osób interesujących się projektowaniem urządzeń cyfrowych. Od czytelnika wymaga się przygotowania matematycznego w zakresie szkoły średniej.

● **Przetwarzanie i przechowywanie informacji w systemach automatyki** — PIEŃKOS J., TURCZYŃSKI J. Wyd. WNT, Warszawa 1974, s. 439, cena zł 40,—
Struktury cyfrowych systemów automatyki: systemy centralnej rejestracji danych i przetwarzania danych, systemy bezpośredniego sterowania cyfrowego, systemy sterowania nadrzędnego, urządzenia sterowania lokalnego. Elementy cyfrowe: elementy logiczne rezystorowo-tranzystorowe RTL — seria LOGISTER E-100 H, elementy logiczne o dużej odporności na działanie zakłóceń — HTL, elementy logiczne tranzystorowe TTL itd. Układy i zespoły funkcjonalne do przetwarzania informacji: liczniki, sumatory, multipleksery, rejestry rozdzielcze, komparatory, metody i układy przetwarzania liczb dziesiętnych na dwójkowe (i odwrotnie), układy pamięci. Układy sprzęgające: układy i urządzenia wejściowe i wyjściowe, układy do transmisji i kontroli informacji cyfrowej. Przykłady rozwiązań cyfrowych systemów automatyki. Dodatki: charakterystyka minikomputerów, podstawowe pojęcia i terminologia dotyczące przetworników A/C i C/A, opis działania logicznego przerzutników. Załączniki: elementy logiczne E-100 H, minikomputery K-202 i MOMIK 8b, pamięć dyskowa PD 9425 itd.
Materiały przeznaczone są dla projektantów komputerów i systemów cyfrowych automatyki.

● **Urządzenia wejścia/wyjścia maszyn cyfrowych** — KOCIATKIEWICZ P., ZAMOJSKI W. — Wyd. Politechniki Wrocławskiej, Wrocław, 1974, s. 230, cena zł 24,— (skrypt).
Urządzenia taśmy papierowej: technika taśmy papierowej, czytnik, dziurkarka, dalekopis, elektryczna maszyna do pisania. Urządzenia kart dziurkowanych: technika karty dziurkowanej, czytniki, dziurkarki, Drukarka wierszowa. Czytniki dokumentów. Monitory ekranowe. Pisaki. Pamięci masowe. Organizacja współpracy między urządzeniami zewnętrznymi a komputerem: zasady współpracy urządzeń zewnętrznych z komputerem, uniwersalne kanały wprowadzania i wyprowadzania informacji, standardowy interface.
Materiały przeznaczone są dla studentów wyższych szkół technicznych.

● **Elementy cyfrowe** — KAPŁON K., KOŁODZIEJ R., KURAS A. — Wyd. Politechniki Wrocławskiej, Wrocław 1974, s. 349, cena zł 36,— (skrypt).
Wprowadzenie. Własności elementów półprzewodnikowych, stosowanych w układach przełączających. Analiza elementów cyfrowych. Synteza elementów cyfrowych. Rezystorowo-tranzystorowe elementy cyfrowe (RTL). Elementy cyfrowe o sprzężeniu bezpośrednim (DCTL). Diodowo-tranzystorowe elementy cyfrowe. Realizacja elementów DTL w technice scalonej. Elementy logiczne TTL. Elementy logiczne z przełączaniem prądowym (ECL). Elementy logiczne z tranzystorami polowymi. Porównanie elementów logicznych scalonych.
Materiały przeznaczone są dla studentów wyższych szkół technicznych.

● **Konstrukcja urządzeń cyfrowych** — JELENIEWSKI T., KOCIATKIEWICZ P. — Wyd. Politechniki Wrocławskiej, Wrocław 1974, s. 296, cena zł 33,— (skrypt).
Uwagi ogólne o urządzeniach cyfrowych. Układy komunikacji bezpośredniej operatora z urządzeniem cyfrowym. Generatory i układy synchronizujące w urządzeniach cyfrowych. Rejestry. Sumatory binarne. Liczniki. Pamięci operacyjne. Pamięci stałe. Zasilanie urządzeń cyfrowych. Problemy zakłóceń w urządzeniach cyfrowych. Zastosowanie maszyn cyfrowych do projektowania urządzeń cyfrowych.
Materiały przeznaczone są dla studentów wyższych szkół technicznych.

● **System kształcenia inżynierów organizatorów-informatyków** — DWORCZYK M. — Red. Wyd. Politechniki Warszawskiej, Warszawa 1974, s. 274, cena zł 19,—
Sylwetka zawodowa inżyniera organizatora informatyka — TROCKI M.
Analiza zapotrzebowania na inżynierów organizatorów informatyków dla przemysłu elektromaszynowego — BOROWSKI J.
Analiza systemu kształcenia inżynierów organizatorów w instytutach inżynieryjno-ekonomicznych ZSRR — ZAWADZKI M.
Analiza systemu kształcenia inżynierów organizatorów w uniwersytetach amerykańskich — DWORCZYK M.
Koncepcja systemu kształcenia i doskonalenia kadry inżynierów organizatorów dla przemysłu elektromaszynowego — DWORCZYK M., ZAWADZKI M.
Założenia zbierania informacji o planach i programach kształcenia inżynierów organizatorów informatyków dla potrzeb kierunku studiów — organizacja i zarządzanie przemysłem w Politechnice Warszawskiej — ZUBER R.
Niniejszą publikację przygotowano na podstawie pracy pt. „Opracowanie projektu kompleksowego kształcenia i doskonalenia kadry inżynierów organizatorów informatyków dla przemysłu elektromaszynowego", wykonanego na zlecenie Ośrodka Badawczo-Rozwojowego Informatyki. Praca zawiera pomocnicze materiały do przedmiotu. Wstęp do specjalizacji wykładanej na pierwszym roku studiów Politechniki Warszawskiej.

● **Analiza liniowych obwodów prądu zmiennego w stanie nieustalonym** (program ANNIE) — GRABOWIECKI T. Wyd. Politechniki Śląskiej im. W. Pstrowskiego, Gliwice 1973 r., str. 106, cena zł 7,—. Seminarium obwodów elektrycznych. Z. 3.
Sposób posługiwania się programem: zastosowanie, standardowy opis gałęzi obwodu, wprowadzanie zasadniczych danych, rodzaje obliczeń itp. Opis programu: czytanie i drukowanie danych, segregowanie gałęzi, tworzenie drzewa normalnego, obliczanie przebiegów czasowych, zakończenie programu itp. Program „ANNIE". Przykłady.
Materiały przeznaczone są dla studentów Wydziału Elektrycznego Politechniki Śląskiej, innych wyższych szkół technicznych oraz inżynierów-projektantów.

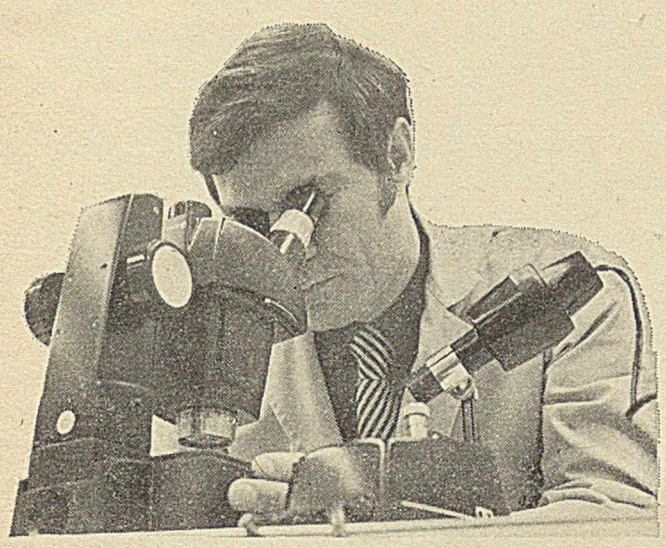
Łatwo było mówić, że jesteśmy najlepsi, gdy — pierwsi i jedyni w świecie — zaczęliśmy produkować komputerowe taśmy magnetyczne.

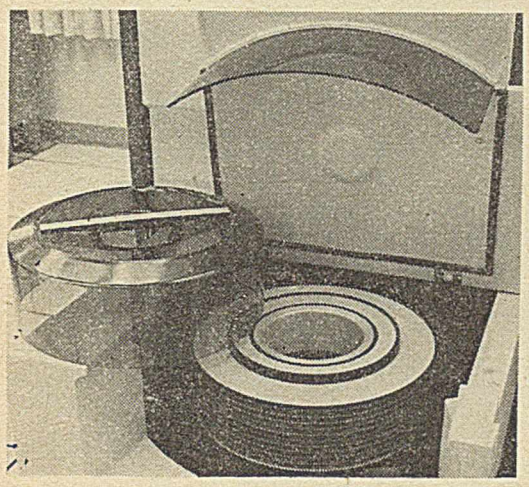
To było 20 lat temu, w 1953 r. specjalnie powleczona taśma acetatowa naszej produkcji zapoczątkowała nową erę elektronicznego przetwarzania danych. Z chwilą ukazania się naszych taśm na rynku producenci komputerów mogli zaprojektować urządzenia do magazynowania danych z nieosiągalną dotąd gęstością zapisu informacji i szybkością ich odszukiwania. 3 M prowadząc intensywne prace badawczo-rozwojowe z myślą o potrzebach przemysłu komputerowego — stale udoskonalało swoje wyroby, ulepszając technologię, zwłaszcza powłok magnetycznych i materiałów podkładowych. Jako pierwsi wprowadzaliśmy kolejno na rynek:

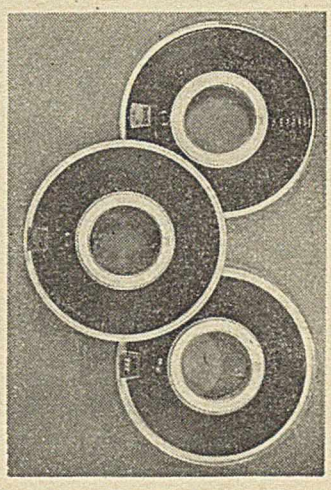
- taśmę na podkładzie poliestrowym
- taśmę z 50 mikrocalową warstwą ochraniającą powłokę tlenkową
- taśmę przystosowaną do trudnych warunków eksploatacyjnych
- taśmę o przedłużonym okresie eksploatacji i dużej odporności na uszkodzenia powłoki tlenkowej („dropout")
- taśmę testowaną wyłącznie aparaturą elektroniczną
- taśmę eliminującą uszkodzenia powodowane nieprawidłową eksploatacją.

Taśmy ze znakiem „Scotch" stały się podstawą ustanowienia międzynarodowych norm dla taśm magnetycznych i — tradycyjnie już — spełniają one wymagania stawiane przez różnorodne konstrukcje sprzętu komputerowego. Produkujemy w oparciu o samodzielnie wytwarzane tlenki, materiały podkładowe oraz własne receptury spoiw.

Nośniki magnetyczne produkujemy w nowocześnie wyposażonych fabrykach, gdzie sprawdzone kwalifikacje personelu technicznego i surowa kontrola produkcji stanowią o najwyższej jakości naszych wyrobów.
Chcemy być nie tylko pierwsi, ale i najlepsi!
A oto nasze najnowsze wyroby:







Komputerowa taśma magnetyczna 3M „Scotch" marki # 700 charakteryzuje się teksturowanym podłożem zabezpieczającym przed zarysowaniem, zakleszczaniem oraz zanieczyszczaniem powierzchni.

Pakiet dyskowy odporny na uszkodzenia („crashgurad") charakteryzuje się unikalnymi własnościami powłok, która chroni przed uszkodzeniami powierzchni magnetycznej wskutek zatarcia głowicą („headcrash"), a w przypadku poważniejszych zatarć, zmniejsza tworzenie się aglomeratów tlenku oraz powstawanie zanieczyszczeń głowic zapisu/odczytu.

3M (EAST) AG
CH 6301 ZUG BAARERSTRASSE 8 SWITZERLAND
TELEPHONE 042 23 21 31 TELEX 78 873 CABLE TRIMINCO ZUG

Cena 8 zł.–

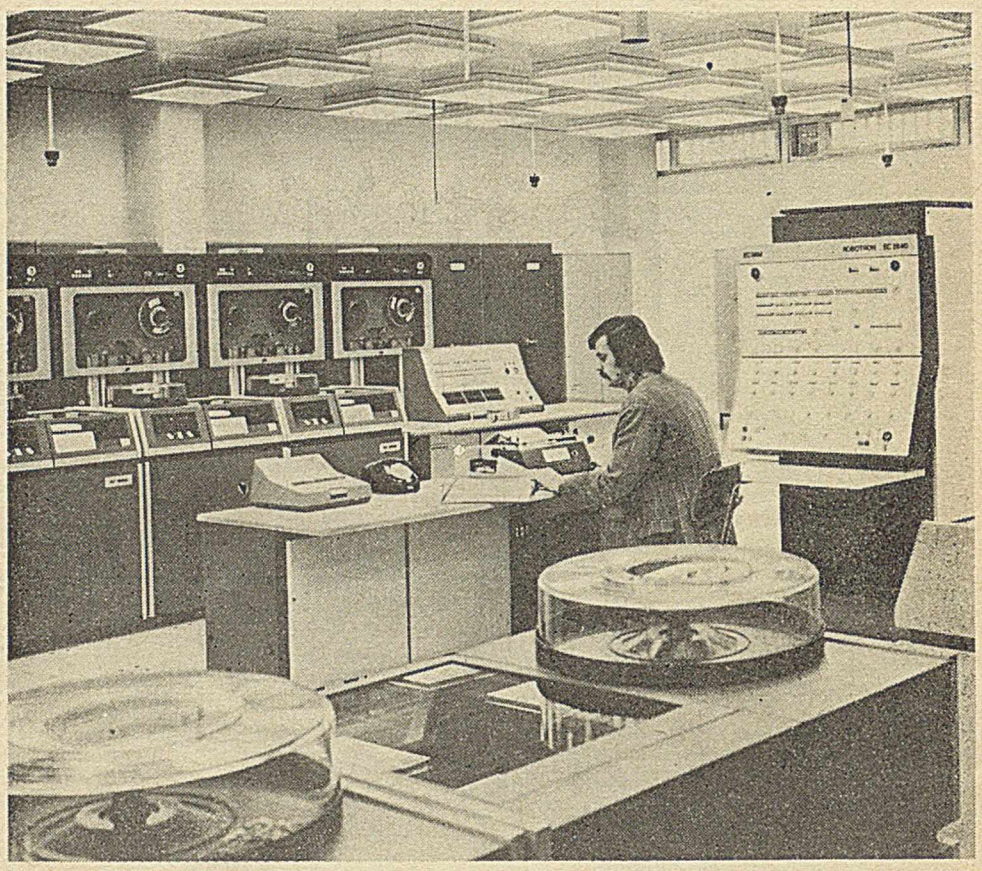
robotron

Problemy
Wymagania
Rozwiązania

stają przed Wami podobnie jak przed nami!

Nasza droga:

JS — jednolity system elektronicznej techniki obliczeniowej. Oznacza to wspólnotę ponad granice krajów — olbrzymi potencjał w służbie producentów i użytkowników.



bme

Eksporter:

Büromaschinen-Export GmbH Berlin NRD — 108 Berlin Fridrichstrasse 61

Przedstawicielstwo w Polsce BME, Biuro Techniczno-Handlowe przy Ambasadzie NRD ul. Filtrowa 62 m 63, 02-057 Warszawa

Sprzedaż i informacje:

PREDOM-Org. ul. Górskiego 9, 00-033 Warszawa MS

Nasza oferta:

Elektroniczna Maszyna Cyfrowa **ROBOTRON JS 1040** ze zmienną liczbą urządzeń zewnętrznych JS do rozwiązywania zadań ekonomicznych i naukowo-technicznych, do zwiększania skuteczności zarządzania, planowania i ewidencji, do zastosowania jako komputer główny sterujący w systemie hierarchicznym. Dokumentacja systemów ukierunkowana maszynowo i problemowo, usługi szkoleniowe dostosowane do konkretnych wymagań użytkownika, wypróbowany serwis techniczny — oto zakres świadczonych przez nas usług.

Nasz program dostaw i usług jest dla Was korzystny. Prosimy o zasięganie bliższych informacji.

WCT/1781/K/74